

STRATEGIES FOR FILTERING INCORRECT MATCHES  
IN SEABED IMAGE MOSAICING

HAMED BAGHERI









STRATEGIES FOR FILTERING INCORRECT MATCHES IN SEABED IMAGE

MOSAICING

by

© Hamed Bagheri

A Thesis submitted to the

School of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Engineering

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

July 2012

St. John's

Newfoundland

## Abstract

Population statistics have a wide range of applications in the fishery industry, oceanographic research (e.g., population studies, habitat analysis), as well as for the oil and gas industry (e.g., population monitoring for environmental impact assessment). Most Remotely Operated Vehicles (ROV) or Autonomous Underwater Vehicles (AUV) image acquisition transects produce overlap between successive or adjacent images, such that individual animals could appear in several images, which could yield inaccurate counts. In order to eliminate the possibility of counting the same animal more than once, the overlap between images must be detected. We developed a feature-based mosaicing algorithm that uses the Scale Invariant Feature Transform (SIFT) in which feature descriptors of images are extracted and appropriate correspondences are found and matched by computing the standardized Euclidean distance between descriptor vectors. We present a new strategy for finding correct correspondences and discarding incorrect matches from the background using spatial clustering and standardized Euclidean distance for computing an adaptive threshold value used by the second-best match method. Results are provided to validate the proof of concept for our strategy.

### Acknowledgements

I would like to acknowledge the advice and guidance of my supervisors Dr. Andrew Vardy and Dr. Ralf Bachmayer for providing me with great opportunities and carefully guiding me during my graduate studies. I also wish to express my sincere gratitude to them for providing me with financial assistance throughout my Master's program.

I am also grateful to Mr. Adam Gobi for his support and feedback.

I acknowledge the Canadian Healthy Oceans Network (CHONe), the Faculty of Engineering and the School of Graduate Studies for their financial and academic support for this project.

I would like to thank my family; my father, Mr. Mostafa Bagheri for his valuable support and guidance throughout my life, and my mother Mrs. Mina Shahbaz for her unconditional love and support. Your constant care, love and attention helped me get through the hardest times of my life. Thank you for always being there for me and constantly encouraging me every step of the way.

Last but certainly not least I would like to thank my friends for supporting and encouraging me to pursue this degree.

Hamed Bagheri

## Table of Contents

Abstract .....	ii
Acknowledgements .....	iii
Table of Contents .....	iv
List of Tables .....	viii
List of Figures .....	ix
List of Abbreviations .....	xvii
Chapter 1 .....	1
Introduction .....	1
1.1 Thesis Contributions .....	5
1.2 Thesis Organization .....	7
Chapter 2 .....	8
Background .....	8
2.1 Seabed Habitat Mapping .....	8
2.2 Literature Review .....	9
2.2.1 Photo-Mosaicing .....	9
2.2.2 Lighting Problem .....	13
2.2.3 Distance Metrics .....	13

2.2.4	Stitching Software.....	14
2.2.5	Seafloor Habitat Mapping.....	15
2.3	Motivation.....	15
Chapter 3.....		17
Underwater Imaging: Common Problems.....		17
3.1	Colour Enhancement.....	17
3.2	Marine Snow.....	18
3.3	Non-uniform lighting.....	19
3.3.1	Histogram Processing.....	20
3.3.2	Homomorphic Filtering.....	24
3.4	Experimental Results.....	27
Chapter 4.....		35
Image Overlap Detection.....		35
4.1	Fourier-based Methods.....	36
4.1.1	Extracting Translation.....	37
4.1.2	Extracting Rotational Degree.....	37
4.1.3	Extracting Scale Ratio.....	38
4.1.4	FFT-based Image Registration for Underwater Images.....	41
4.2	Feature Based Method.....	42

4.2.2	Feature Matching .....	50
Chapter 5 .....		68
Image Registration .....		68
5.1	Image Transformation Models .....	68
5.1.1	Rigid-Body Transformation (Isometric Transformation) .....	69
5.1.2	Similarity Transformation .....	69
5.1.3	Affine Transformation .....	70
5.1.4	Projective Transformation or Homographies .....	70
5.1.5	Perspective Projection .....	71
5.2	Geometric Model Estimation .....	74
Chapter 6 .....		79
Image Blending .....		79
Chapter 7 .....		83
Results .....		83
7.1	Distance Metrics Comparison .....	83
7.2	Results of Feature Matching .....	88
7.3	Final Image Mosaics .....	89
7.4	Discussion .....	103
Chapter 8 .....		105



Conclusion .....	105
8.1 Future work .....	105
Bibliography .....	107
Appendix A .....	114

## List of Tables

Table 5.1: Number of required iterations for geometric model estimation. ....	77
Table 7.1: Distance metrics and threshold value comparison #1.....	85
Table 7.2: Distance metrics and threshold value comparison #2.....	86
Table 7.3: Distance metrics and threshold value comparison #3.....	87
Table 7.4: Parameters of the implemented mosaicing algorithm.....	92
Table 7.5: Adaptive thresholds computed by our proposed method for image set #1.....	102
Table A.1: List of adaptive thresholds computed for image set #2. ....	121
Table A.2: List of adaptive thresholds computed for image set #3. ....	128
Table A.3: List of adaptive thresholds computed for image set #4. ....	135
Table A.4: List of adaptive thresholds computed for image set #5. ....	142
Table A.5: List of adaptive thresholds computed for image set #6. ....	149

## List of Figures

Figure 1.1: Typical Remotely Operated Vehicle (ROV) tracks for collecting imagery data. a) Lawn mower pattern b) Dense center; the dark line illustrates the trajectory of the submersible and the shaded area shows the region imaged by the vehicle's camera. ....	2
Figure 1.2: Overlap between sequential image frames. ....	2
Figure 1.3: Illustrating two captured images with animals appearing in the overlap of images causing the multiple counting problem. ....	3
Figure 1.4: Image pair illustrating the multiple counting problem for a captured scene of starfish. ....	4
Figure 1.5: Diagram showing the counting steps procedure. ....	6
Figure 3.1: Dominant blue and green colour in underwater imagery caused by light absorption effect. ....	18
Figure 3.2: Marine snow; bright floating particles in these images are considered as noise. .....	19
Figure 3.3: Images with non-uniform illumination [33]. ....	19
Figure 3.4: The original grayscale underwater image and its histogram. ....	21
Figure 3.5: Equalized image and its corresponding histogram. ....	22
Figure 3.6: The original grayscale images and its histogram before applying CLAHE. ....	23
Figure 3.7: Image and its histogram after applying CLAHE with <i>clip limit</i> = 0.2. ....	24
Figure 3.8: Diagram of Homomorphic filtering. ....	25
Figure 3.9: Cross section of a circularly symmetric filter function. ....	26

Figure 3.10: Cross section of the designed Homomorphic filter .....	28
Figure 3.11: The original grayscale image.....	29
Figure 3.12: Image after applying CLAHE .....	29
Figure 3.13: Image after applying the Homomorphic filter.....	30
Figure 3.14: The original grayscale image showing marine snow .....	32
Figure 3.15: Illustrating the effect of CLAHE on marine snow .....	32
Figure 3.16: Result of applying Homomorphic filtering to marine snow.....	33
Figure 4.1: Image pair used for FFT registration.....	40
Figure 4.2: Registered images with FFT-based method. Extracted $scale=1.21$ , $rotation=24.66$ degrees, translation $(x, y) = (-229, 246)$ .....	40
Figure 4.3: Mosaic of a pair of images with translation and projective effect registered by FFT-based method .....	42
Figure 4.4: Different scales of the blurred images and computation of DoGs are shown. Local extrema are then detected [34].....	45
Figure 4.5: Detection of maxima and minima of DoG with neighbourhood pixels [34]....	46
Figure 4.6: Forming the SIFT feature descriptors. This figure illustrates a 2x2 descriptor formed from an 8x8 sample array [34]. ....	47
Figure 4.7: Image captured with arbitrary viewpoint. 3 SIFT keypoints are shown. ....	48
Figure 4.8: Image captured with another different viewpoint showing 3 extracted SIFT keypoints .....	48
Figure 4.9: SIFT feature descriptors; 4x4 SIFT descriptor matched in two different images of the same scene .....	49

Figure 4.10: This matrix illustration shows a sample distance computation between descriptor of image $I_1$ and all $N_2$ descriptors of $I_2$ ; both are 128 dimension vectors.....	54
Figure 4.11: Illustrating indices of the closest and the second closest distances.....	54
Figure 4.12: Illustrating features with multiple correspondences. <i>SIFT Threshold = 0.66</i> .....	56
Figure 4.13: Multiple correspondence problem corrected. <i>SIFT Threshold = 0.66</i> .....	56
Figure 4.14: Illustrating an example situation where correspondences are considered as correct matches. Dots show feature points in each left and right images and circles illustrate clusters of feature points. ....	65
Figure 4.15: Features of one cluster in the left image have correspondences in several different feature point clusters in the right image; in this situation correspondent pairs are discarded. ....	65
Figure 4.16: Keypoints of one cluster in $I_2$ are associated with two clusters in $I_1$ . This is the condition where the correspondences are rejected. Yellow arrows show the direction of two red lines.....	66
Figure 4.17: Illustrating the condition where correspondences are accepted. Yellow arrows show three lines between matched keypoints between a pair of clusters. ....	67
Figure 5.1: Submersibles pose: illustrating translation and rotation parameters affecting image frames and the overlap area.....	73
Figure 5.2: RANSAC family in 3 broad categories [54]. ....	75
Figure 6.1: Multiband blending algorithm adds Laplacian pyramids of blending images and then reconstructs the seamless output image.....	81

Figure 6.2: An image divided into two different intensity levels in order to generate a sample sharp edge between images. ....	82
Figure 6.3: A slight boundary is visible after applying the multi-band blending method. ....	82
Figure 7.1: Sample image pair 1. The selected area shows the same region captured in two different photos. ....	84
Figure 7.2: Sample image pair 2. The circled area shows the same region in two different photos. ....	86
Figure 7.3: Sample image pair 3. The circled area shows the same region in two different photos. ....	87
Figure 7.4: The conventional matching method, [34], with $threshold=0.8$ . Resulting in 37 correct matches and 110 incorrect matches making up to 25% accuracy. ....	89
Figure 7.5: Feature matching using the proposed strategy. Up to 84% accuracy with 32 correct matches and 6 incorrect matches. ....	89
Figure 7.6: Mosaic created by stitching images with the multi-band blending function. The rockfish circled on the top was originally located on the boundary of one image; now it is clearly visible once in the image mosaic. ....	90
Figure 7.7: Mosaic of two images of starfish illustrating smooth boundaries multi-band blending algorithm. The starfish on top of the image is captured in two different photos appearing with minimum artifacts in the mosaic. ....	91
Figure 7.8: Image set #1, including images {1, 2, 3, 4, 5, 6}. Images are collected for mosaicing purpose by the U.S. Geological Survey. ....	93
Figure 7.9: Feature matching between images {1} on the left and {2} on the right. ....	94

Figure 7.10: Images stitched $\{1\} \leftarrow \{2\}$ .....	94
Figure 7.11: Feature matching between images $\{1, 2\}$ on the left and $\{3\}$ on the right. ....	95
Figure 7.12: Images stitched $\{1, 2\} \leftarrow \{3\}$ .....	95
Figure 7.13: Feature matching between images $\{1, 2, 3\}$ on the left and $\{4\}$ on the right. ....	96
Figure 7.14: Images stitched $\{1, 2, 3\} \leftarrow \{4\}$ .....	97
Figure 7.15: Feature matching images $\{1, 2, 3, 4\}$ on the left and $\{5\}$ on the right. ....	98
Figure 7.16: Images stitched $\{1, 2, 3, 4\} \leftarrow \{5\}$ .....	99
Figure 7.17: Feature matching between images $\{1, 2, 3, 4, 5\}$ on the left and $\{6\}$ on the right. ....	100
Figure 7.18: Images stitched $\{1, 2, 3, 4, 5\} \leftarrow \{6\}$ . A photomosaic of 6 images from the sea floor collected by USGS. ....	101
Figure A.1: Image set #2, including images $\{6, 7, 8, 9, 10, 11\}$ . ....	115
Figure A.2: Feature matching between images $\{6\}$ on the left and $\{7\}$ on the right. ....	116
Figure A.3: Images stitched $\{6\} \leftarrow \{7\}$ .....	116
Figure A.4: Feature matching between images $\{6, 7\}$ on the left and $\{8\}$ on the right. ....	117
Figure A.5: Images stitched $\{6, 7\} \leftarrow \{8\}$ .....	117
Figure A.6: Feature matching between images $\{6, 7, 8\}$ on the left and $\{9\}$ on the right. ....	118
Figure A.7: Images stitched $\{6, 7, 8\} \leftarrow \{9\}$ .....	118
Figure A.8: Feature matching between images $\{6, 7, 8, 9\}$ on the left and $\{10\}$ on the right. ....	119
Figure A.9: Images stitched $\{6, 7, 8, 9\} \leftarrow \{10\}$ .....	119

Figure A.10: Feature matching between images {6, 7, 8, 9, 10} on the left and {11} on the right. ....	120
Figure A.11: Images stitched {6, 7, 8, 9, 10} $\leftarrow$ {11} .....	120
Figure A.12: Image set #3, including image {11, 12, 13, 14, 15, 16}.....	122
Figure A.13: Feature matching between images {11} on the left and {12} on the right. ....	123
Figure A.14: Images stitched {11} $\leftarrow$ {12} .....	123
Figure A.15: Feature matching between images {11, 12} on the left and {13} on the right. ....	124
Figure A.16: Images stitched {11, 12} $\leftarrow$ {13} .....	124
Figure A.17: Feature matching between images {11, 12, 13} on the left and {14} on the right. ....	125
Figure A.18: Images stitched {11, 12, 13} $\leftarrow$ {14} .....	125
Figure A.19: Feature matching between images {11, 12, 13, 14} on the left and {15} on the right. ....	126
Figure A.20: Images stitched {11, 12, 13, 14} $\leftarrow$ {15} .....	126
Figure A.21: Feature matching between images {11, 12, 13, 14, 15} on the left and {16} on the right. ....	127
Figure A.22: Images stitched {11, 12, 13, 14, 15} $\leftarrow$ {16} .....	127
Figure A.23: Image set #4, including images {16, 17, 17, 19, 20, 21}. ....	129
Figure A.24: Feature matching between images {16} on the left and {17} on the right. ....	130
Figure A.25: Images stitched {16} $\leftarrow$ {17} .....	130



Figure A.26: Feature matching between images {16, 17} on the left and {18} on the right. .....	131
Figure A.27: Images stitched {16,17} $\leftarrow$ {18} .....	131
Figure A.28: Feature matching between images {16, 17, 18} on the left and {19} on the right. ....	132
Figure A.29: Images stitched {16,17,18} $\leftarrow$ {19} . ....	132
Figure A.30: Feature matching between images {16, 17, 18, 19} on the left and {20} on the right. ....	133
Figure A.31: Images stitched {16,17,18,19} $\leftarrow$ {20} .....	133
Figure A.32: Feature matching between images {16, 17, 18, 19, 20} on the left and {21} on the right. ....	134
Figure A.33: Images stitched {16,17,18,19,20} $\leftarrow$ {21} .....	134
Figure A.34: Image set #5, including images {21, 22, 23, 24, 25, 26} . ....	136
Figure A.35: Feature matching between images {21} on the left and {22} on the right. ....	137
Figure A. 36: Images stitched {21} $\leftarrow$ {22} .....	137
Figure A.37: Feature matching between images {21, 22} on the left and {23} on the right. .....	138
Figure A.38: Images stitched {21,22} $\leftarrow$ {23} .....	138
Figure A.39: Feature matching between images {21, 22, 23} on the left and {24} on the right. ....	139
Figure A.40: Images stitched {21, 22, 23} $\leftarrow$ {24} . ....	139

Figure A.41: Feature matching between images {21, 22, 23, 24} on the left and {25} on the right. ....	140
Figure A.42: Images stitched {21, 22, 23, 24} $\leftarrow$ {25} .....	140
Figure A.43: Feature matching between images {21, 22, 23, 24, 25} on the left and {26} on the right. ....	141
Figure A.44: Images stitched {21, 22, 23, 24, 25} $\leftarrow$ {26} . ....	141
Figure A.45: Image set #6, including images {26, 27, 28, 29, 30, 31} . ....	143
Figure A.46: Feature matching between images {26} on the left and {27} on the right. ....	144
Figure A.47: Images stitched {26} $\leftarrow$ {27} .....	144
Figure A.48: Feature matching between images {26, 27} on the left and {28} on the right. ....	145
Figure A.49: Images stitched {26, 27} $\leftarrow$ {28} .....	145
Figure A.50: Feature matching between images {26, 27, 28} on the left and {29} on the right. ....	146
Figure A.51: Images stitched {26, 27, 28} $\leftarrow$ {29} .....	146
Figure A.52: Feature matching between images {26, 27, 28, 29} on the left and {30} on the right. ....	147
Figure A.53: Images stitched {26, 27, 28, 29} $\leftarrow$ {30} .....	147
Figure A.54: Feature matching between images {26, 27, 28, 29, 30} on the left and {31} on the right. ....	148
Figure A.55: Images stitched {26, 27, 28, 29, 30} $\leftarrow$ {31} .....	148

## List of Abbreviations

AHE	Adaptive Histogram Equalization
AUV	Autonomous Unmanned Vehicle
CDF	Cumulative Distribution Function
CLAHE	Contrast Limited Adaptive Histogram Equalization
DoG	Difference-of-Gaussian
FFT	Fast Fourier Transform
GPS	Global Positioning System
HIS	Hue, Saturation and Intensity
NN	Nearest Neighbour
RGB	Red Green Blue
ROPOS	Remotely Operated Platform for Ocean Sciences
ROV	Remotely Operated Vehicles
SIFT	Scale Invariant Feature Transformation
SVD	Singular Value Decomposition
USGS	U.S. Geological Survey

# Chapter 1

## Introduction

Monitoring the benthic habitat of marine environments has wide application in the oil and gas industries (e.g., population monitoring for environmental impact assessment), as well as for oceanographic research (e.g., population studies, habitat analysis) [1]. In order to use these imagery data effectively, there is a need to develop means to extract information from raw imagery. With few exceptions, this step has been done manually until recently [2], [1] and [3], where the researchers count the number of animals seen in images or video sequences for further study. Assuming the automatic or the manual counting works properly and effectively, there are still conditions that will make the multiple counting of an animal probable, resulting in inaccurate statistics. For example, an animal might be counted several times if it appears in multiple images or several times in a video stream. A common scenario for this problem can be demonstrated as follows: A submersible which is used for exploring the sea floor could follow either of the tracks showing in Figure 1.1; case (a): for some sections of the tracks there is a possibility of overlapping regions with adjacent tracks. In case (b), the circled area shows an area which is explored multiple times as it is chosen to be the starting point for several data collecting explorations.

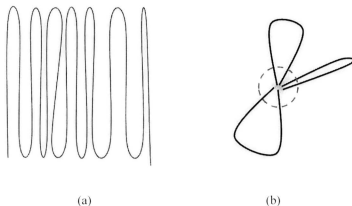


Figure 1.1: Typical Remotely Operated Vehicle (ROV) tracks for collecting imagery data.  
a) Lawn mower pattern b) Dense center; the dark line illustrates the trajectory of the submersible and the shaded area shows the region imaged by the vehicle's camera.

Not only could the overlap between adjacent or crossing tracks cause a counting problem, but also in a normal condition where the submersible is following a route, image frames can also have overlap. If an animal appears in the image boundaries, it might also be counted several times. This situation is illustrated in Figure 1.2.

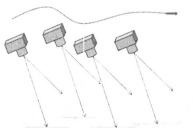


Figure 1.2: Overlap between sequential image frames.

Figure 1.3 and Figure 1.4 will illustrate the situation we are referring to as the multiple counting problem. In the image pair depicted in Figure 1.3, a total number of four

individual crabs exist on the seafloor; whereas, the appearance of two crabs in two different images might cause multiple counting of the animals which results in six crabs in the scene.

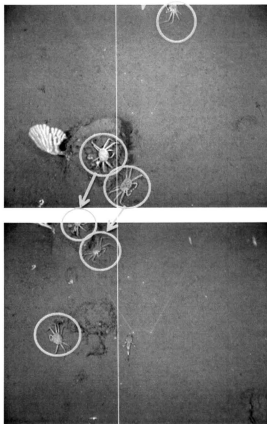


Figure 1.3: Illustrating two captured images with animals appearing in the overlap of images causing the multiple counting problem.

In Figure 1.4, a bed of starfish is captured by the camera. As can be seen by a simple inspection, except for one starfish which is not captured in the first frame but only in the second frame, the other four starfish are present in both image frames.

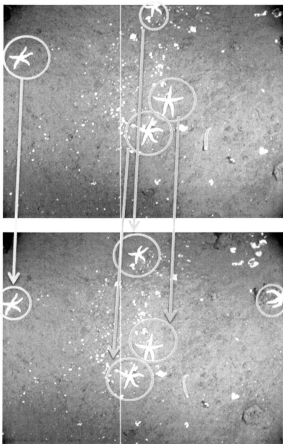


Figure 1.4: Image pair illustrating the multiple counting problem for a captured scene of starfish.

## 1.1 Thesis Contributions

In this work, a collection of image processing algorithms are gathered in a unique way to design strategies tolerant to a large number of incorrect feature matches. Producing a system able to tolerate projective distortion, as well as low contrast images for the image stitching purpose is another goal of this research. However, the aim of this research is not to generate a map of the explored seafloor area. The final goal is to generate only one representation of a captured scene in the processed image dataset to prevent potential multiple counting of animals.

Moreover, a novel approach is proposed for estimating a previously pre-defined parameter in feature matching. This threshold value has been fine-tuned manually.

The imagery data being used for this work were taken from the Barkley canyon plate off the west coast of Vancouver Island, BC, Canada [4] and the U.S. Geological Survey (USGS) imagery dataset [5]. The Barkley canyon imagery is challenging since it suffers from lack of sufficient overlap between images. Moreover, low contrast, non-uniform lighting, blurry images due to the water quality and the non-planar view in some images can also be listed as issues. Accurate georeferenced data of the images were not provided either. Therefore, in order to match images, the scope was limited to using only the images without additional metadata. For generating a larger mosaic a set of images with sufficient overlap should have been used. For this purpose the USGS dataset has been used to justify and demonstrate the robustness of the proposed strategy.

Currently, images captured by ROVs are pre-analysed by human operators. The steps are illustrated in Figure 1.5. Our focus is on the pre-analysis step.





Figure 1.5: Diagram showing the counting steps procedure.

On the seafloor, moving species can be categorized in three groups. Organisms such as corals are sessile creatures and are immobile [6]. Sea stars include a very large variety of species living on the sea floor all over the world. These animals propel slowly along the fine-sediment seafloor [6]. Rockfish are animals living on the ocean floor having significant movements [7]. Throughout this work, we are assuming animals are not moving significantly between successive images.

This research has resulted in the following publications and presentations:

- H.Bagheri, A.Vardy, and R. Bachmayer, “*Strategies for Filtering Incorrect Matches in Seabed Mosaicing*”, has been accepted for inclusion in the Proceedings OCEANS’11 MTS/IEEE KONA, Hawaii, September 2011.
- H.Bagheri, A.Vardy, and R. Bachmayer, “*Creating Seabed Image Mosaics for Counting Benthic Species*”, Presentation in workshop on underwater robotics, Memorial University, St. John’s, NL, May 2011.
- H.Bagheri, A.Vardy, and R. Bachmayer, “*Image Mosaicing for the multiple counting problem in benthic habitat mapping*”, Abstract accepted in MeshAtlantic Video Survey Techniques Workshop, Faro University of Algarve/CCMAR, June 2011.

- H.Bagheri, A.Vardy, and R. Bachmayer, “*Image Mosaicing for Benthic Species Multiple Counting Problem*”, Poster presentation at CHONe annual conference, Montreal, May 2011.
- H.Bagheri, A.Vardy, and R. Bachmayer, “*Image Mosaicing for Benthic Species Counting*” in Proc. IEEE NECEC 2011.St.John’s, NL.

## 1.2 Thesis Organization

This thesis is organized as follows. Chapter 2 gives a comprehensive literature review and our motivation for conducting this research. Chapter 3 addresses the issue of underwater imaging and illumination constancy for underwater image processing. Chapter 4 focuses on image overlay detection methods, and presents an in-depth analysis of the feature-based method including feature extraction, feature matching, and clustering. The image registration and geometric model estimation are discussed in Chapter 5 followed by a description of the image blending algorithm used in the mosaicing system in Chapter 6. Chapter 7 illustrates and summarizes the results with a section on future work concluding this thesis. An appendix is provided to include more image mosaics to justify the proposed strategy.

# Chapter 2

## Background

In this chapter, background information about seafloor animals' habitat mapping and a detailed literature review of relevant research in similar areas will be outlined. At the end of this chapter, the motivation for conducting this research will be presented.

### **2.1 Seabed Habitat Mapping**

Canada has the largest shoreline in the world which includes fifteen distinctly different marine ecosystems [8]. Dealing with issues such as how species are related to the specification of their habitat and evaluating biodiversity in these ecosystems is essential for managing oceans resources in a sustainable manner. It has been recognized that characterizing the relationship between biodiversity and habitat in the Arctic due to significant changes in the polar environment is one of the most urgent needs regarding ocean health in Canada [8]. Monitoring seafloor organisms using underwater images is an important tool for scientists to work toward a better understanding. Therefore, developing image analysis and object recognition tools for quantifying the abundance and diversity of the different seabed organisms is highly desirable.

## 2.2 Literature Review

The presented work involved research along several different directions. This review provides a detailed view of existing approaches in the field of photo-mosaicing. Identifying similar features between images is an important computer vision problem with applications in numerous fields, such as image stitching, object detection, image registration, object localization, object recognition, image retrieval and mapping. This procedure remains challenging due to the existing problems such as partial occlusion of objects, illumination changes and camera viewpoint changes [9].

### 2.2.1 Photo-Mosaicing

The field of image mosaicing is relatively old with an extensive research literature. Photo mosaicing methods in the research literature are mainly categorized in two groups, i.e., direct methods ( [10], [11] and [12]) and feature-based methods ( [13], [14], [15]and [16]). Direct methods use all the available image data and can provide accurate results, but are heavily sensitive to image ‘brightness constancy’ i.e., the tendency for an object to be perceived as having the same brightness under varying lighting conditions, as well as initialization due to executing iterative algorithms [17]. On the other hand, the feature based methods use special characteristics of an image such as the corners. Recently developed feature based methods use invariant features which makes the mosaicing system more robust to light changes, camera motion and pose. The most recent work on feature extraction has focused on local invariant features [18], with applications such as image stitching [19], 3D modeling, gesture recognition, object recognition [19] and

robotic mapping [20]. Affine invariant and scale invariant features are presented in [21]. This matching algorithm is able to achieve affine invariance with an 80 degree change of camera angle. This is done by introducing two camera orientation parameters named camera longitude angle and camera latitude angle. The latitude angle in this case is similar to the tilt. This method simulates possible view changes in affine space to gain high matching accuracy. Szeliski and Shum [22], presented an approach for creating panoramic mosaics from image sequences. They aim for not having any constraints on how the images are taken or how camera motion should be controlled. The 3D rotations are recovered directly in their method instead of using 8-parameter planar perspective transforms. However, in this work it is assumed that the camera is centered at the origin. In general, an approximately planar scene is defined where the ratio between the distance to the scene and the variation in ground elevation is high. In [23], Marks *et. al.* introduce a real-time system for video mosaicing the ocean floor. Their approach uses visual correspondences during both acquisition and mosaicing to ensure there are no gaps in the mosaic. In their work, problems such as camera field of view under perspective and orthographic projections, as illustrated in Figure 2.1, as well as gaps between the captured images are discussed.



Figure 2.1: (a) Perspective projection: results of different observations of the same scene in the overlapping part. (b) Orthographic projection: cameras observe the same scene in the overlapping area.

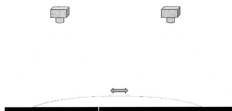


Figure 2.2: A gap between captured images on the top of the knoll. Part of the scene is out of the camera field of view.

Photo mosaicing is affected by several issues. The most important ones are addressed in this work, and are related to projection geometry, lighting problems, camera orientation and non-static scenery or non-planar factors such as fish and their shadows. In this work a downward pointing camera mounted on an ROV was used. In another attempt [24], a projective transformation formula for compensating the perspective distortion in the overlapping area of two images is used. Szeliski, discussed in [11], proposed an arithmetic-based method for generating a photomosaic of colour images. Occlusion in photo mosaicing is of great significance. This issue is addressed in [25], where an effort was made to reduce occlusion and distortion caused by trees in ortho-image production.

Pritchett and Zisserman in [26], presented an approach which generates sets of local planar homographies. These sets are to be used for providing a more robust affinity measure for potentially matching correspondences and to restrict the searching stage for potential feature matches. Defining *wide baseline* as the condition where the distance between the cameras compared to the viewed scene is large, Deriche in [27] proposed a general method for matching images from an uncalibrated camera for short baseline stereo matching. In their paper, correlation is used over a sized search window for matching two features in a pair of images. A. Baumberg in [28] presents an automatic feature matching method for images seen from two arbitrary viewpoints. In his work, unlike previous stereo matching methods, no prior information about the relative camera position and orientation were assumed. In fact, one of the goals of this work was to determine this information from image feature matches. Their approach extends the rotation invariants method with local affine image transformations. Y. Cao *et. al.* in [29] proposed a two-level matching strategy composed of Affine-SIFT and colour moments invariants. Their strategy is to work with the fusional feature which speeds up the process of finding the local correspondences. Also some works use Global Positioning System (GPS) data for image matching. The authors in [30], present a prototype of a system for image based localization in urban environments by using images tagged with GPS locations. Their work is based on a wide baseline matching system using scale invariant features to select the best image match from the database.

### 2.2.2 Lighting Problem

Several techniques have been proposed for solving the lighting problem in underwater images. In order to compensate for the true colour problem, colour contrast is equalized in Red Green Blue (RGB) colour space followed by transforming the image into Hue, Saturation and Intensity (HSI) and then stretching the saturation and intensity of the image [31]. In [32], eliminating the crinkle pattern by choosing appropriate tiles in Contrast Limited Histogram Equalization (CLAHE) is discussed, which is implemented in the MATLAB Image Processing Toolbox as well [32]. For compensating for the effect of the non-uniform lighting problem, Local Histogram Equalization and Homomorphic filtering are used in [33].

### 2.2.3 Distance Metrics

Several scholars have conducted research on methods for improving SIFT feature matching. In D. Lowe's method [34] and [35], the best candidate match is found by identifying the nearest neighbour in the database of the keypoints with a minimum Euclidean distance. In [36], the metric Euclidean distance is replaced with a combination of city block distance and chessboard distance where, despite the improvement in efficiency, this algorithm suffers from using two randomly defined thresholds. These pre-defined variables make the algorithm unsuitable for images from a larger database with different characteristics. A. Baumberg employed the Mahalanobis distance metric to measure the similarity between features in [28]. The same metric is used by [29] where the Mahalanobis distance between two moment invariants with a predefined threshold along



with the normalized correlation between the corresponding regions is used for the measure of similarity. Ferrer *et al* in [37] presented a technique for creating a photo mosaic using navigational data for underwater images. Their work was an effort to design a system for creating mosaics with the available sparse positions and orientation information.

The common goal of many computer vision algorithms is to extract geometric information from image data. Due to errors in matching, the available data is usually corrupted with a large number of incorrect correspondences. In [38], the authors addressed this problem as they proposed a data points classification method by using the generated hypothesis directly so that the need for pre-defining an inlier threshold is eliminated.

#### **2.2.4 Stitching Software**

In this study several available software programs were also tested for the Barkley Canyon imagery data set. The AutoStitch is the implementation of [19] by M. Brown and D. Lowe. This software creates a panorama from the input images. Hugin is the implementation of [39] by P. d'Angelo. Radial light falloff as well as exposure variation, white balance variation and non-linear camera response from overlapping images are the main foci in their work. An online demonstration of Affine SIFT [21] is also available for testing the effectiveness of the method on different image pairs.

### **2.2.5 Seafloor Habitat Mapping**

With the exception of a few papers ([1], [3] and [2]), we are not aware of research on automating population counting of animals for any purpose.

## **2.3 Motivation**

Finding the overlay of images has been an interesting problem in image processing and computer vision. This process can also take advantage of other available information about the position of the camera or the viewed scene specifications. For example, knowing the positioning information of the aerial images will assist the algorithm to decide if a pair of images is taken from nearby locations or not and they will be considered as separate images if they are taken from far enough away. A similar approach is valid for underwater images as well, where the positioning data is available.

Another possible situation is when the trajectory of the camera is known. In this case, it will be possible to guess the direction of the sequences and further computations can utilize this available information. Some other factors can affect the decision making on which an overlay detection algorithm is to be used. For example, availability of the required information for camera calibration, information about the installation of the camera on the submersible, distances between laser pointers and the camera, or the camera viewpoint angle all have an important role in mosaicing.

The imagery data used for this work was collected for the purpose of finding corals on the Barkley Canyon plate off the west coast of Vancouver Island, BC, Canada. These images were not taken for the purpose of mosaicing; therefore the overlap of images was not an

issue at the time. Moreover, no accurate information about the location of the photos was available; therefore, techniques for grouping the images to categorize nearby images for further computation were not possible. Not having the camera information led us to estimate the perspective geometric transformation between images rather than using projective transformation. Moreover, the underwater scenery is challenging, with common underwater imaging problems such as the lighting problem resulting in low quality images.

These limitations have motivated us to propose strategies for matching non-distinctive image features, as well as for repeated objects in the images to be used by further mosaicing systems.

# Chapter 3

## Underwater Imaging: Common Problems

One of the major problems for processing underwater images is related to the effect of light in the aquatic environment. Light quality suffers two different processes, namely absorption and scattering. The former is where light disappears from the process, and the latter describes the direction of photons, which is mainly caused by different sizes of particles in the water. These processes cause some unwanted effects in underwater images, such as the blurring effect. In the past few years, conducted research shows that aquatic environments raise new challenges due to the light effects. In this chapter, common problems in underwater image processing will be briefly addressed. We also discuss the effect of non-homogeneous lighting in section 3.3.

### 3.1 Colour Enhancement

In underwater situations, the light absorption effect causes one colour to overshadow other colours in the image. In this environment, colours fade one by one depending on their wavelength. Green and blue colours travel the furthest in the water due to their short wavelength. This effect results in images with high blue or green colour density. An example of this problem is illustrated in Figure 3.1.

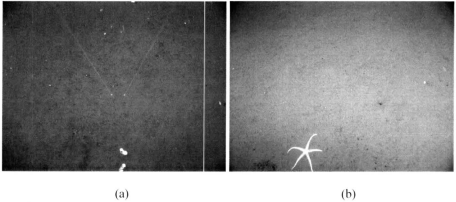


Figure 3.1: Dominant blue and green colour in underwater imagery caused by light absorption effect.

Details about solving problems of this type are discussed in [40]. In their method, a twofold approach is proposed in which contrast stretching on the RGB colour space is performed to equalize the image colour contrast then HSI colour space is used to increase the true colour by stretching the saturation and intensity.

### 3.2 Marine Snow

Marine snow refers to particles composed of dead materials and organisms slowly drifting from higher levels of the ocean. These particles floating in the distance between the camera and the seabed reflect the light carried by the vehicle causing very bright particles in the scene. This can be considered to be noise in underwater imagery as can be seen in Figure 3.2.

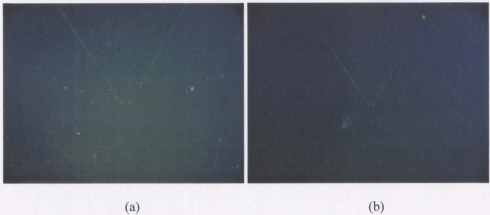


Figure 3.2: Marine snow; bright floating particles in these images are considered as noise.

### 3.3 Non-uniform lighting

In the deep sea environment, natural light is insufficient on the seafloor and submersibles have to carry their own lighting source to provide adequate lighting. Artificial lights illuminate the scene in a non-uniform fashion, in which there is primarily a bright spot at the center of the image, and the surrounding area is poorly illuminated, as shown in Figure 3.3.

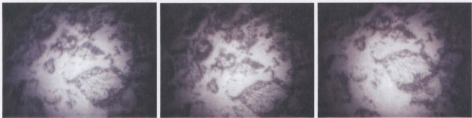


Figure 3.3: Images with non-uniform illumination [33].

In this section the lighting effect and illumination model are reviewed. Histogram specification and Homomorphic filtering are introduced and compared for how they

compensate for the lighting problem. The experimental results of the techniques mentioned are presented in chapter 6 and the conclusion closes this discussion. More details about illumination model, spatial and frequency domain filtering are available in [41].

### 3.3.1 Histogram Processing

Images with a limited range of grayscale suffer from lack of contrast. To enhance the quality of the images, spatial domain techniques can be employed. The term spatial domain refers to the image and manipulation of pixels in the image in which operations will take place on an origin pixel and neighbourhood pixels around the origin. In this section three of these methods will be briefly discussed and compared.

#### 3.3.1.1 Histogram Equalization

In Histogram Equalization (HE) the image contrast is maximized by mapping the image histogram based on the probability distribution of the grayscale image.

This method consists of four stages:

---

#### **Histogram Equalization**

---

1. Creating the image histogram.
  2. Calculating the Cumulative Distribution Function (CDF) of the histogram.
  3. Calculating the new values of the histogram.
  4. Assigning new values for each gray-level in the image.
-

The CDF function is defined as equation 3.1.

$$CDF(k) = \sum_{i=1}^k Hist(i), \quad (3.1)$$

where  $k$  is a gray-level and  $Hist$  denotes the original image histogram.

The new values for the equalized histogram are calculated by using equation 3.2.

$$EqHist(i) = \frac{CDF(i) - CDF_{min}}{Row \times Col - CDF_{min}} \times (G - 1), \quad (3.2)$$

where  $EqHist(i)$  is the new equalized value of  $i^{th}$  gray-level,  $G$  is the number of gray-levels in the image, and  $Row$  and  $Col$  are the number of rows and columns of the input image respectively.  $CDF_{min}$  is also the minimum value of the calculated CDF.

In the following example, the histogram of the given image is shown and equalized. Although the resulting image may not look constant, the cumulative histogram is an exact linear ramp that indicates that the density is equalized.

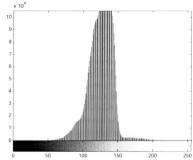


Figure 3.4: The original grayscale underwater image and its histogram.



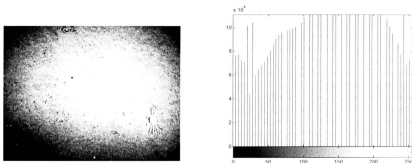


Figure 3.5: Equalized image and its corresponding histogram.

The cumulative graph shows the linear nature of gray-level frequencies within the image [42]. As can be seen, the output of histogram equalization results in an image with improved contrast.

### 3.3.1.2 Adaptive Histogram Equalization

Adaptive Histogram Equalization (AHE) is a more advanced version of histogram equalization. In underwater imaging, the non-uniform nature of light treats different areas of the image differently. For this reason, some authors suggest compensating for the effect of non-uniform lighting by using local histogram equalization [33]. In this method, a histogram is built for each pixel in the image, using a specified number of  $n \times n$  pixel windows but uniquely modifying the central point of the neighbourhood. This operation will take place for each pixel of the image and the result will be a more balanced image [33]. In other words, this method applies HE for smaller windows on the image.

### 3.3.1.3 Contrast Limited Adaptive Histogram Equalization

Contrast Limited Adaptive Histogram Equalization (CLAHE) is an effective algorithm to obtain an enhanced image directly from a raw image without a level adjustment. CLAHE was originally developed for medical imaging. This algorithm operates on small regions of the image and applies the HE to each one. This will enhance the contrast of each region and thus makes hidden features of the image more visible [41]. CLAHE is an improved version of AHE. Noise can be reduced while maintaining the high spatial frequency content of the image by applying a combination of CLAHE, median filtering and edge sharpening. This technique subdivides the image into  $n \times m$  pixel blocks and calculates the histogram of each block. Each window is then equalized by choosing the monotonically non-decreasing gray-level transformation, mapping the histogram of the desired distribution. However, selection of a clipping level limits the enhancement of each block. Those pixel values that exceed the clip limit will be uniformly redistributed across the histogram [33].

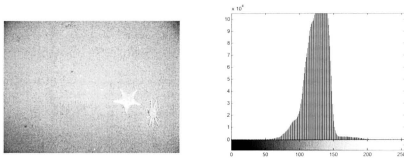


Figure 3.6: The original grayscale images and its histogram before applying CLAHE.

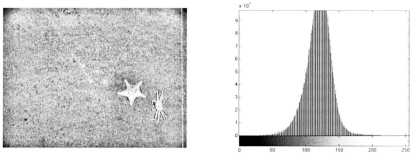


Figure 3.7: Image and its histogram after applying CLAHE with *clip limit*=0.2.

Local HE is computationally expensive and it amplifies noise in low contrast areas [33], whereas CLAHE can help reduce the noise amplification in the low contrast areas of the image.

### 3.3.2 Homomorphic Filtering

A common technique that takes into account the illumination-reflectance model is Homomorphic filtering, which is discussed in detail by R. Gonzales in [41]. In the illumination-reflectance model, the image is represented as a function of illumination and reflectance components. This model is defined in equation 3.3.

$$f(x, y) = i(x, y)r(x, y), \quad (3.3)$$

where  $f(x, y)$  is the image sensed by the camera,  $i(x, y)$  is the illumination and  $r(x, y)$  is the reflectance component. The illumination component of an image is associated with low frequencies in the image which represent slow spatial variations, whereas the reflectance component varies rapidly and is associated with high frequencies. The idea behind this method is to separate the components so that a filter function  $H(u, v)$  can be

applied to each frequency domain separately. On the other hand, by transforming the spatial image to the Fourier domain, separation of the frequency components will not be possible since the Fourier transform converts the multiplication operation into convolution.

The Homomorphic filtering proceeds as follows:

---

### **Homomorphic Filtering**

---

1. Natural logarithm of the grayscale image is taken. This process will convert the multiplication operation between illumination and reflectance to addition.
  2. Resultant image from the previous step is converted to the Fourier domain.
  3. Then the filter function  $H(u, v)$  is applied to the previous output. This filter affects the low and high frequencies of the Fourier transform differently.
  4. Filtered image is now converted back into the logarithm space by taking the inverse Fourier transform.
  5. The final image is achieved by applying the exponential function to the image obtained from the previous step.
- 

The foregoing process is illustrated in Figure 3.8.



Figure 3.8: Diagram of Homomorphic filtering.

The filter response  $H(u, v)$  can be approximated using an ideal high-pass filter. For example, the following formula defines a 2D Gaussian high-pass filter, in which the cutoff frequency is located at a distance  $D_0$  from the origin:

$$H(u, v) = (\gamma_H - \gamma_L) [1 - e^{-cD^2(u, v)/D_0^2}] + \gamma_L, \quad (3.4)$$

where

$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}. \quad (3.5)$$

For an  $M \times N$  size image,  $c$  is the constant to control the sharpness of the slope of the filter function as it transitions from  $L$  to  $H$  [2]. A reduction of dynamic range in brightness with an increase in contrast is expected from this type of filtering. Figure 3.9 illustrates the cross section of  $H(u, v)$ .

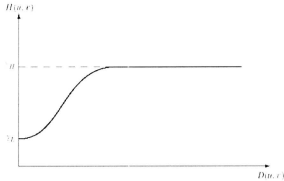


Figure 3.9: Cross section of a circularly symmetric filter function.

If the parameters  $L$  and  $H$  are chosen as  $\gamma_L < 1$  and  $\gamma_H > 1$ , the filter will increase the contribution made by reflectance (high frequencies) and decrease the contribution made

by illumination (low frequencies). The overall result will be simultaneous dynamic range compression and enhancement in contrast.

As a conclusion, it is fair to say that in HE the goal is to enhance the image to gain an optimal overall contrast. However, since underwater images suffer from lack of uniform illumination, it is more suitable to apply local equalization to the images to gain a better result. Homomorphic filtering not only attenuates non-uniform illumination, but also enhances the high frequencies and sharpens the edges of the objects in the image. Results of this discussion can be seen and compared in section 6.

### **3.4 Experimental Results**

In this section the aforementioned filtering algorithms will be applied to several sets of images and the results will be compared. Several tests have been performed to show the effectiveness of the discussed technique in compensating for non-uniform illumination underwater images. The Gaussian high-pass filter used for Homomorphic filtering is defined in equations 3.4 and 3.5. In this type of filtering several parameters have to be selected manually to obtain the enhanced image. For these sets, the Homomorphic filter parameters were chosen as the specification shown in Figure 3.10.

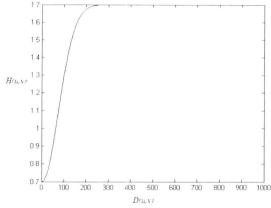


Figure 3.10: Cross section of the designed Homomorphic filter

$$\gamma_L = 0.7$$

$$\gamma_H = 1.7$$

$$c = 0.6$$

$$D_0 = 80$$

The cutoff frequency is located at a distance  $D_0$  from the origin. Parameter  $c$  is the constant to control the sharpness of the slope of the filter function for an  $M \times N$  size image as it transitions from  $\gamma_L$  to  $\gamma_H$ . These parameters should be chosen manually according to the imagery data set. Intensity of light in the center of an image along with the radius of the bright center will give us an approximation to find suitable parameters manually and experimentally. Thus for different lighting intensity these parameters should be modified [1]. Figure 3.11 to Figure 3.13 illustrate images filtered by Homomorphic filtering and CLAHE.



Figure 3.11: The original grayscale image

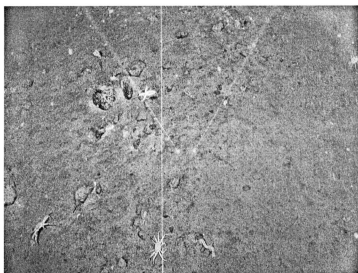


Figure 3.12: Image after applying CLAHE



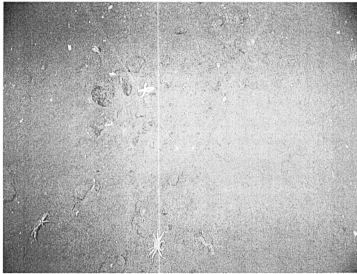


Figure 3.13: Image after applying the Homomorphic filter

As can be seen, CLAHE brings the detail out to the front and brightens the image. On the other hand, some over exposed effects are also visible, e.g., on the starfish and the crab in the bottom of the images. The result of Homomorphic filtering shows an image in which not only is the lighting problem partially solved, but also the edges are sharpened. Comparing the star fish between two images we will see that the result of Homomorphic filtering shows clearer objects with sharper edges.

By definition, an image artifact is any image feature that appears in the image which is not present in the originally captured image. By comparing the edges of objects in the images, e.g., around the legs of the bottom crab, it can be seen that Homomorphic filtering generates a result with fewer image artifacts. Finally, enhancing the visibility of

local details and contrast of the image by CLAHE compared with Homomorphic filtering should not be neglected.

Highlighting all the details in an image is not always desirable. For example, in Figure 3.14 to Figure 3.16 a common problem in underwater images is shown, called marine snow, which is the small particles visible on the top right of each image. These particles are primarily organic detritus and fine-grained sediment continuously falling from the upper layer of the ocean. If we consider marine snow as a type of noise for our image quality, applying CLAHE to the images will increase this noise as it highlights the details. By comparing the original images with the output of CLAHE, marine snow particles look larger in size.

The same discussion is valid for the rest of images with a similar appearance. For example, a sediment pattern on the seafloor with its high self-similarity will appear sharper in images processed by CLAHE. This in turn will generate more similar image feature descriptors which will result in generating more incorrect correspondences. On the other hand, Homomorphic filtering does not increase the marine snow effect.



Figure 3.14: The original grayscale image showing marine snow

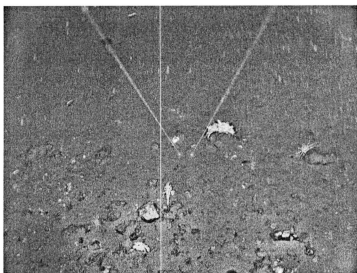


Figure 3.15: Illustrating the effect of CLAHE on marine snow

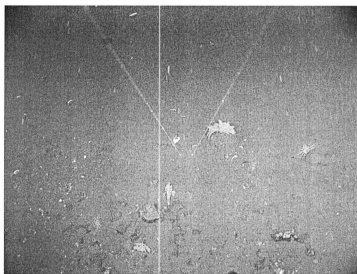


Figure 3.16: Result of applying Homomorphic filtering to marine snow

Figure 3.14 to Figure 3.16 distinguish the functionality of CLAHE and Homomorphic filtering in the case of the existence of marine snow. CLAHE strongly increases the so-called noise while the result of Homomorphic filtering is more realistic. Also, comparing some features in these two images, we can see the effectiveness of Homomorphic filtering. For example, with a closer look at the crab on the bottom left of the image, it can be seen that the crab has become hidden and barely visible by CLAHE whereas Homomorphic filtering makes the crab more visible. This is because Homomorphic filtering corrects the lighting effect on the one hand and sharpened the edges on the other hand. However, in situations where the goal is to make the details and features more visible, CLAHE will be a better choice.

In this section, several image sets were shown and the effect of CLAHE and Homomorphic filtering were discussed. Our conclusion is, depending on the features in the images and what is expected from the process, either CLAHE or Homomorphic filtering can be employed to correct for the lighting effect. In Homomorphic filtering high frequencies that contain interesting imagery data are separated from lower frequencies that contain shading and lighting components. This method not only attenuates the non-uniform illumination but also enhances the high frequencies and sharpens the edges of the objects in the image.

# Chapter 4

## Image Overlap Detection

Image registration is the process of aligning two or more images taken from different viewpoints looking at the same scene in order to align images accurately.

Registration methods can be divided into four broad categories [10]:

- **Pixel based methods:**

Cross-correlation is the basic statistical technique in image registration. This similarity metric is computed from window pairs of a template image and reference images. The cross-correlation based registration can be applied in situations where a slight rotation and scale change exist [43].

- **Fourier method (Phase-correlation):**

Fourier-based registration is more suitable compared to the cross-correlation method in situations where an acceleration of the computational speed is also demanded. Moreover, the phase-correlation method is more suitable if the images are corrupted by noise. This method is based on the Fourier shift theorem.

- **Feature-based methods:**

Feature-based methods use image features derived by a feature extraction algorithm rather than using the pixels' intensity values. The purpose of feature extraction is to filter out redundant information from the original image and to

create abstract information of the image data. Computing a proper geometric transformation relies on the precise selection of these features. These methods are the primary approach for registering two images with an unknown transformation; that is, where the transformation cannot be easily categorized as a rigid-body movement [43] [44].

- **Image registration based on high level features:**

This method uses high level statistical features such as contours and objects for matching images. The drawback of this method is being manual and therefore slow [44]. Moreover, not all the underwater photos captured from the seafloor contain objects which are suitable for contour or boundary recognition.

In this work, the Fourier method and Feature based methods as two widely used categories will be addressed and their functionality for underwater images will be compared.

#### **4.1 Fourier-based Methods**

The Fast Fourier Transform (FFT) based method searches for the optimal match based on the information in the frequency domain [13]. This method is basically implementation of the translation property of the Fourier transform, also known as the Fourier shift theorem. Extracting parameters such as translation, rotation and the scale change between a pair of images of the same scene are discussed as follows:

#### 4.1.1 Extracting Translation

Let  $i_1$  and  $i_2$  be a pair of images with displacement  $(x_0, y_0)$ ,

$$i_2(x, y) = i_1(x - x_0, y - y_0). \quad (4.1)$$

By applying the Fourier transform, the corresponding  $I_1$  and  $I_2$  will be:

$$I_2(\zeta, \eta) = e^{-j2\pi(\zeta x_0 + \eta y_0)} * I_1(\zeta, \eta). \quad (4.2)$$

The phase difference between the images will be calculated by the cross-power spectrum of the two images:

$$\frac{I(\zeta, \eta)I^*(\zeta, \eta)}{|I(\zeta, \eta)I^*(\zeta, \eta)|} = e^{j2\pi(\zeta x_0 + \eta y_0)}, \quad (4.3)$$

where  $I^*$  denotes the complex conjugate of  $I$ . By taking the inverse Fourier transform of equation 4.3, we will have an impulse at the displacement, which is used for optimally registering the two images.

#### 4.1.2 Extracting Rotational Degree

In case there is a translation  $(x_0, y_0)$  and rotation  $\theta_0$  between  $i_2(x, y)$  and  $i_1(x, y)$ , then  $i_1$  and  $i_2$  are related by equation 4.4:

$$i_2(x, y) = i_1(x \cos \theta_0 + y \sin \theta_0 - x_0, -x \sin \theta_0 + y \cos \theta_0 - y_0). \quad (4.4)$$

By taking the Fourier transform,

$$I_2(\zeta, \eta) = e^{-j2\pi(\zeta x_0 + \eta y_0)} \times I_1(\eta \cos \theta_0 + \zeta \sin \theta_0, -\zeta \sin \theta_0 + \eta \cos \theta_0). \quad (4.5)$$

Let  $M_1$  and  $M_2$  be the magnitude of  $I_1$  and  $I_2$ ,



$$M_2(\zeta, \eta) = M_1(\zeta \cos \theta_0 + \eta \sin \theta_0, -\zeta \sin \theta_0 + \eta \cos \theta_0). \quad (4.6)$$

By considering the magnitude of  $I_1$  and  $I_2$ , we can see that the magnitude of  $M_2$  is a rotated version of the magnitude of  $M_1$ .

In order to deduce this rotation, translational displacement is replaced with polar coordinates i.e.,

$$M_1(\rho, \theta) = M_2(\rho, \theta - \theta_0). \quad (4.7)$$

By using phase correlation, as discussed previously, angle  $\theta_0$  will be found.

#### 4.1.3 Extracting Scale Ratio

By extending this theory to the case where  $i_1$  is a scaled replica of  $i_2$  with scale factors  $(a, b)$  for the horizontal and vertical direction and with translation and rotation,

$$i_2(x, y) = i_1(ax, ay). \quad (4.8)$$

By using the Fourier shift theorem, we can solve for the case where  $i_1$  is the scaled, rotated and translated version of  $i_2$ .

$$I_2(\zeta, \eta) = \frac{1}{|ab|} I_1\left(\frac{\zeta}{a}, \frac{\eta}{b}\right). \quad (4.9)$$

Scaling can be reduced to translation by converting the axis to logarithmic scale, i.e.,

$$I_2(\log \zeta, \log \eta) = I_1(\log \zeta - \log a, \log \eta - \log b). \quad (4.10)$$

This is similar to the form:

$$I_2(x, y) = I_1(x - c, y - d), \quad (4.11)$$

where  $y = \log \eta$ ,  $x = \log \zeta$ ,  $c = \log a$  and  $d = \log b$ .

The translation  $(c, d)$  can be computed by the phase correlation technique and the scaling  $(a, b)$  can be found from the translation  $(c, d)$  denoted as  $a = e^c$  and  $b = e^d$ , where  $e$  is the natural logarithm. By changing the scale from  $(x, y)$  to  $\left(\frac{x}{a}, \frac{y}{b}\right)$  their polar representation will be:

$$\begin{aligned}
 \rho_1 &= (x^2 + y^2)^{1/2}, \\
 \theta_1 &= \tan^{-1}\left(\frac{y}{x}\right), \\
 \rho_2 &= \left(\left(\frac{x}{a}\right)^2 + \left(\frac{y}{a}\right)^2\right)^{1/2} = \left(\frac{1}{a}\right)(x^2 + y^2) = \frac{\rho_1}{a}, \\
 \theta_2 &= \tan^{-1}\left(\frac{x/a}{y/a}\right) = \tan^{-1}\left(\frac{x}{y}\right) = \theta_1.
 \end{aligned} \tag{4.12}$$

If  $i_1$  is the scaled, translated, and rotated version of  $i_2$ , their Fourier magnitude spectra in polar representation are shown as:

$$\begin{aligned}
 M_1(\rho, \theta) &= M_2\left(\frac{\rho}{a}, \theta - \theta_0\right), \\
 M_1(\log \rho, \theta) &= M_2(\log \rho - \log a, \theta - \theta_0),
 \end{aligned} \tag{4.13}$$

which can be written as equation 4.14.

$$M_1(\zeta, \theta) = M_2(\zeta - d, \theta - \theta_0), \tag{4.14}$$

where

$$\begin{aligned}
 \zeta &= \log \rho, \\
 d &= \log a.
 \end{aligned} \tag{4.15}$$

Then by using the phase correlation formula, angle  $\theta_0$  and scale  $a$  will be computed. As an example, Figure 4.1 illustrates an image pair with translation, scale and rotation. The registered image is shown in Figure 4.2 using FFT-based registration.

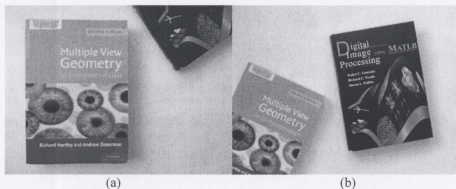


Figure 4.1: Image pair used for FFT registration.

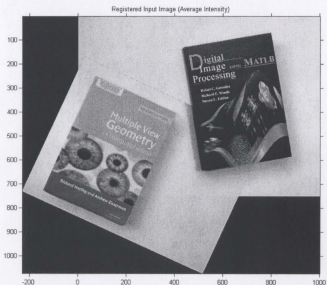


Figure 4.2: Registered images with FFT-based method. Extracted  $scale=1.21$ ,  $rotation=24.66$  degrees, translation  $(x, y) = (-229, 246)$ .

In the example illustrated in Figure 4.2, averaging the grayscale values of pixels is used as the image blending method. As can be seen, the boundaries of images in the overlap area are clearly visible as sharp edges. In chapter 6 we will also present a multi-band blending method in which the sharp boundaries in the overlapping area of the image mosaics will be converted to seamless transition of the grayscale values.

The image pair shown in Figure 4.1 is taken by a downward looking camera perpendicular to the image plane. In cases where the camera is not perpendicular to the image, the FFT-based image registration method may not be accurate. According to the presented algorithm, this method is able to extract one degree of image rotation only. In the actual ROV images, the camera is not necessarily looking downward in most cases. Therefore, this method should be replaced with a method able to tolerate different camera view angles. Section 4.1.4 presents an example of FFT-based registration for a pair of images taken by an ROV for the purpose of clarity.

#### **4.1.4 FFT-based Image Registration for Underwater Images**

By using FFT-based image registrations, our aim is to highlight and compare the importance of the camera viewpoint in a dataset. As discussed in section 4.1.2, this registration method can extract only one rotational degree of the camera viewpoint.

Figure 4.3 illustrates an example of this method on an actual image of the seafloor taken by an ROV.

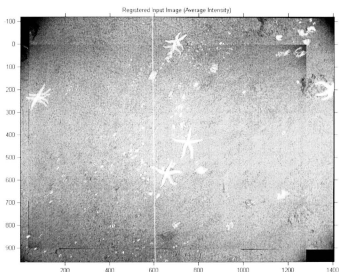


Figure 4.3: Mosaic of a pair of images with translation and projective effect registered by FFT-based method

As can be seen, scale, translation and one degree of rotation are taken into account. On the other hand, due to the 3D projection effect, two images cannot be fully registered, yielding a blurred mosaic. For example, the starfish on the top and left, since they appear on the edge of the image, are affected most by the projective effect. According to this discussion, not knowing the relative geometry of the seafloor to the camera or the unknown camera view angles encourages us to assess other suitable registration methods.

## 4.2 Feature Based Method

In the context of local invariant features, for any object in an image, the features represent interesting points of the object, ranging from complex features such as the object itself to simpler structures such as edges or points. Also, these features can be designed to be

invariant to scale and orientation, and to be robust to changes in viewpoint, illumination, noise, and blurring.

Robust and accurate feature matching can be achieved by extracting the more invariant features. Therefore, discriminative features should not be extracted from image intensity or colour values in an image due to inconsistency of illumination between images.

Region features are generally high-contrast closed-boundary regions marked by their centre of gravity [45]. These features are invariant to scaling, skewing, rotation and image intensity variation. Regions of interests are identified by segmentation procedure. These groups of features can find large rotations, scale changes and translations.

Line features are another category of feature extractors. These algorithms are suitable for identifying contours. Popular edge detectors such as the Canny, Harris or Laplacian filter are included in this category. Region feature extractors are generally more robust compared to line feature extractors [46]. A survey of performance evaluations of edge detection techniques can also be found in [46].

The most widely used image registration methods are based on feature extractors using features based on point localization. These point region feature extractors can provide descriptors which correspond to the feature point coordinate.

Feature extraction algorithms which rely on the first derivative analysis such as Harris are more robust and less sensitive to variation of noise compared to algorithms using the second derivative or Gaussian curvature. An extensive survey of the performance of local descriptors is presented in [47], in which it is discussed that embedding a descriptor into

the features keypoints will enhance the process of finding correct feature correspondences.

To the best of our knowledge, the most widely-used algorithm that incorporates all the aforementioned advantages is the Scale Invariant Feature Transform (SIFT), [34].

#### 4.2.1.1 SIFT Feature Extraction

SIFT is known as a robust feature extraction method published by D. Lowe [34]. This algorithm is invariant to changes in rotation, translation and scale, and is partially invariant to changes in 3D transformation (viewpoint) and illumination. SIFT describes and detects local features and possesses the particular characteristics of invariance and robustness.

There are four detection stages for SIFT features. The first stage is scale-space extrema detection which involves applying the Gaussian function in order to blur the image.

$$L(x, y, \sigma) = G(x, y, \sigma) * i(x, y), \quad (4.16)$$

where  $i(x, y)$  is the input image,  $L(x, y, \sigma)$  is the Gaussian-blurred image, and the Gaussian function  $G$  with kernel size based on  $\sigma$  is defined as:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-((x^2 + y^2)/2\sigma^2)}, \quad (4.17)$$

The difference of Gaussian space is formed by convolving an image with a Difference-of-Gaussian filter (DoG):

$$G(x, y, k\sigma) - G(x, y, \sigma), \quad (4.18)$$

i.e.,

$$\begin{aligned}
 D(x, y, \sigma) &= G(x, y, k\sigma) * i(x, y) - G(x, y, \sigma) * i(x, y) \\
 &= L(x, y, k\sigma) - L(x, y, \sigma),
 \end{aligned}
 \tag{4.19}$$

where  $k$  is a constant coefficient.

This is basically the difference of the blurred images with Gaussian filters at scale  $\sigma$  and  $k\sigma$  as shown in Figure 4.4.

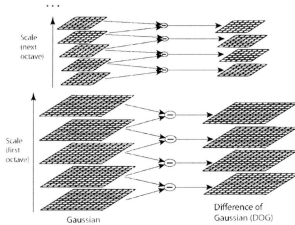


Figure 4.4: Different scales of the blurred images and computation of DoGs are shown. Local extrema are then detected [34].

DoG space of images is then obtained and grouped by applying  $G * i$  with increasing  $\sigma$  by octave. An octave corresponds to a doubling of  $\sigma$ . The interest points (called keypoints herein) are then identified as the local minima or maxima of the DoG space. Each pixel in the DoG space is then compared to its eight neighbours in the same scale and all the eighteen neighbours in the higher and lower scales. If the pixel is a maximum or minimum among all the neighbour pixels, it is identified as a keypoint, as illustrated in Figure 4.5.



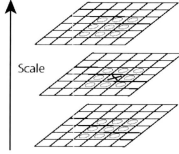


Figure 4.5: Detection of maxima and minima of DoG with neighbourhood pixels [34]

The second stage of SIFT is keypoint localization. In this stage, the position of each keypoint candidate is determined using interpolation of nearby data. Also, keypoints with low contrast or those that are classified as belonging to edges are rejected.

In the third stage, an orientation is assigned to the keypoints. To compute the orientation, by using the Gaussian smoothed image,  $L$ , at the closest scale to the candidate keypoints scale, a gradient orientation histogram is computed in the neighbourhood of the keypoint. Each neighbour pixel is weighted by the gradient magnitude and a Gaussian window with equal to 1.5 times the scale of the keypoint. The image  $L(x, y)$  with the closest value of  $\sigma$  is used for computing the gradient magnitude and orientation by the following equations:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}, \quad (4.20)$$

$$\rho(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))). \quad (4.21)$$

The fourth and final stage of SIFT is the formation of the feature descriptors. The feature descriptor is computed as a group of histograms of a quadratic pixel neighbours. The contribution of each pixel is weighted by the gradient magnitude and by a Gaussian filter with  $\sigma$  equal to 1.5 times the scale of the keypoint. According to this, the vector with respect to  $\rho(x, y)$  is then stored so that the descriptor vector is invariant to rotation. Each of these histograms contains eight bins, and each descriptor includes an array of the histograms around the keypoint as shown in Figure 4.6. This means that the SIFT feature descriptor has  $4 \times 4 \times 8 = 128$  elements. That is  $4 \times 4$  sub region histograms containing 8 bins each. In order to enhance the invariance to changes in illumination, the descriptor vector is then normalized to unit length.

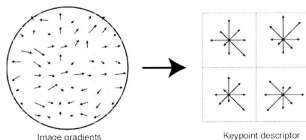


Figure 4.6: Forming the SIFT feature descriptors. This figure illustrates a  $2 \times 2$  descriptor formed from an  $8 \times 8$  sample array [34].

Some examples of SIFT keypoints extracted from a pair of images are illustrated in Figure 4.7 and Figure 4.8. These images are captured with an arbitrary viewpoint angle, scale and translation.



Figure 4.7: Image captured with arbitrary viewpoint. 3 SIFT keypoints are shown.



Figure 4.8: Image captured with another different viewpoint showing 3 extracted SIFT keypoints.

For the purpose of illustration, three matched SIFT keypoints are shown in each image. It can be clearly observed that subsets of the extracted keypoints remain the same. For the

purpose of comparison, two of the matched keypoints are shown in Figure 4.9 in a larger scale.

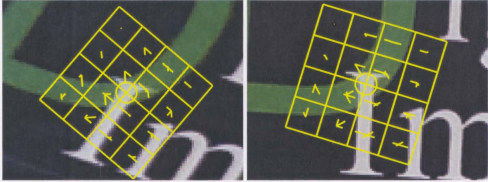


Figure 4.9: SIFT feature descriptors; 4x4 SIFT descriptor matched in two different images of the same scene

An array of histograms around the keypoint is shown in Figure 4.9. Each of these bins contains 8 orientations for a  $4 \times 4$  histogram array resulting in a  $8 \times 4 \times 4 = 128$  dimension descriptor vector. The actual size of each spatial bin is  $k\sigma$  where  $k$  is a nominal factor and  $\sigma$  is the scale of the keypoint.

#### 4.2.1.1.1 Affine-invariant SIFT

In ROV imagery data, objects may appear in images with significantly different viewpoints. These images captured in varying viewpoints undergo 3D deformations. Affine-transforms of the image plane can approximate these deformations. An affine transformation is an invertible transformation which is a composition of rotations, translations, dilations and shears. Using homogeneous coordinates, this transformation can be shown as equation 4.22. Homogeneous coordinates represent a 2D vector in 3D space so that translation can be included in matrix manipulation.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x r_{11} & r_{12} & t_x \\ r_{21} & s_y r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (4.22)$$

where  $(x, y)$  and  $(x', y')$  are pixel coordinates in the images. The parameters  $r, t$  and  $s$  represent the rotation, translation and scale for mapping the pixel  $(x, y)$  to the pixel  $(x', y')$ . The SIFT method is successful in being fully invariant to four parameters out of six of an affine transform, namely scale, rotation and translation invariant.

The method illustrated by Affine-SIFT basically simulates different camera viewpoints, namely latitude and longitude angles, and uses SIFT for extracting features. This method is mathematically proven to be fully affine invariant as discussed in [21], but the simulation of possible viewpoints will be an expensive process therefore SIFT is selected as the algorithm of choice for the rest of this research.

#### 4.2.2 Feature Matching

Identifying similar features between images is an important computer vision problem with application in numerous fields, such as image stitching, object detection, image registration, object localization, object recognition, image retrieval and mapping. This procedure remains challenging due to existing problems such as partial occlusion of objects, illumination changes, and camera viewpoint changes.

The problem being addressed in this section is finding a reliable set of feature matches given two arbitrary images of a scene or object. In this chapter, several distance metrics along with an ambiguity measure are studied and compared to measure similarity between

feature vectors. The experimental results show that using the standardized Euclidean distance can increase the ratio of inliers over total matches.

#### 4.2.2.1 The Conventional Feature Matching Method

In the conventional feature matching method [34], the best candidate match for each keypoint is found by the nearest neighbour of the SIFT features. For finding the nearest neighbours, the minimum Euclidean distance for each candidate matching pair is computed. However, many features will not be correctly matched by this technique. Therefore, it is useful to employ a method to discard incorrect correspondences. A global threshold on the Euclidean distance between the descriptor vectors does not perform well enough as some features are more discriminative than others. In order to make this process more robust to potential wrong matches, a ratio of the nearest neighbour to the second nearest neighbour is computed. Candidates with a ratio of less than a pre-defined threshold value are chosen as matching correspondences. We will refer to this method as *the second-best match method*. This method performs well as discussed in [17], and the idea behind this method is that the correct matches need to be significantly closer than the closest incorrect match to be reliable. In the original work [34], this threshold value is chosen equal to 0.8. Defining  $Dist_1$  and  $Dist_2$  as the first and the second best match distances corresponding to a feature descriptor in the dataset, a match is accepted if equation 4.23 is satisfied.

$$\frac{Dist_1}{Dist_2} < 0.8. \quad (4.23)$$

Notice the distance ratio has a low value where the best match is at a significantly closer distance compared to the second best match. In contrast, a high value of the distance ratio is obtained in a condition where the feature point has at least one strong competitor in terms of distance. In a discussion presented in [34], it is mentioned that this threshold value is able to eliminate 90% of the false matches while it rejects 5% of the correct matches [34].

#### **4.2.2.2 The Proposed Feature Matching Method**

Many of the initial correspondences may become incorrect matches due to the ambiguous and non-distinctive features from the background. The background clutter in our case is the sediment pattern. This pattern generates highly similar feature descriptors with inter-feature distances within a small range. In other words, due to the high dimensionality of the feature space, it is highly probable that a large number of correspondences are within very similar distances to each other. The second-closest matching method can reject a significant percentage of false matches. However, having a pre-defined threshold value can highly affect our decision making strategy for the next stage of processing, which is image registration. This means, if we have a suitable percentage of correct matches, using a geometry model fitting algorithm such as Random Sample Consensus (RANSAC), [48], the algorithm will be able to estimate the transformation parameters. However, if there is an insufficient percentage of correct matches among incorrect matches, additional strategies should be used to increase this percentage, or more sophisticated algorithms

compared to RANSAC should be employed. These are the trade-offs that we need to manage.

Tables 7.1 to 7.3 in the result section will show how a fixed threshold value can influence the number of correct matches; and moreover, slight changes in the threshold value can yield different results in finding matched feature descriptors. These tables also highlight the disadvantages of using a pre-defined threshold value that we attempt to resolve by defining an adaptive thresholding method.

#### 4.2.2.2.1 Multiple Correspondences

Multiple correspondences refer to the situation where several keypoints in an image are associated with one single keypoint in another image. This problem could occur where the variation of distances between features descriptors is small. In the original algorithm for finding matching candidates, a situation might occur where multiple descriptors of an image are chosen as matches for one single descriptor of a secondary image. Here, we are trying to find common features of image  $I_1$  and  $I_2$  in which  $N_1$  features are extracted from  $I_1$  and  $N_2$  features from  $I_2$ . To find a suitable matching candidate for the  $i^{\text{th}}$  descriptor of  $I_1$ , the distance between  $i^{\text{th}}$  descriptor of  $I_1$  and all  $N_2$  descriptors of  $I_2$  are computed as illustrated in Figure 4.10.



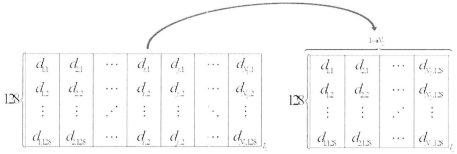


Figure 4.10: This matrix illustration shows a sample distance computation between descriptor of image  $I_1$  and all  $N_2$  descriptors of  $I_2$ ; both are 128 dimension vectors.

In this illustration,  $d_{i,j}$  is  $j^{\text{th}}$  element of  $i^{\text{th}}$  SIFT feature descriptor in which  $j < 128$  and  $i < \text{maximum number of keypoints}$ .

Based on the second-best match method, feature  $j^{\text{th}}$  of  $I_2$  is chosen as a match with  $i^{\text{th}}$  feature of  $I_1$  if the following condition defined in equation 4.24 is satisfied.

$$\frac{Dist_{\min}}{Dist_{2^{\text{nd}}-\min}} < Threshold \quad (4.24)$$

The closest distance and the second closest distance are shown by  $Dist_{\min}$  and  $Dist_{2^{\text{nd}}-\min}$ .

Indices of the descriptors are also shown in Figure 4.11 as  $Idx_m$  and  $Idx_n$  for descriptor  $m$  and  $n$  from  $I_2$ .

$Dist(I, N_2)$		$N_2$ distances	
$Dist_{\min}$	$Dist_{2^{\text{nd}}-\min}$	...	
$Idx_m$	$Idx_n$	...	

Figure 4.11: Illustrating indices of the closest and the second closest distances

In a case where the features are non-distinctive and are very similar to each other, the following problem could occur, except that in this case descriptors  $m$  and  $o$  from  $I_2$  have the shortest distances with a descriptor  $j$  from  $I_1$ . If the condition shown in equation 4.24 is satisfied, features  $j$  and  $m$  will be selected as a candidate pair. This is where the multiple matching problem occurs, i.e., when feature  $m$  is associated with more than one feature of  $I_1$ . The proposed method for solving this problem is described as follows:

For features of the second image with multiple matches, a reverse match finding stage will be performed. This means we try to find a match for the mentioned feature among all the features of the first image by using the same algorithm. If the result of the two stages has common members, the common correspondences will be chosen as a match.

Another possible solution for multiple matching of SIFT descriptors could be that the nearest match among the multiple matches is chosen as the correct correspondence. This is because, throughout the process, we witness the previously mentioned method performs better than the latter one.

#### 4.2.2.2.2 Results of Resolving the Multiple Correspondences

For the purpose of illustration an example of this situation is circled in Figure 4.12. As can be seen, for a small threshold value such as 0.66 there are multiple matches for the starfish and the snail shell in the bottom left of  $I_2$ . This shows there are several numbers of feature pairs for which their second-best match threshold is less than 0.66. Figure 4.12

and Figure 4.13 illustrate an example of this situation before and after investigating for multiple matches.

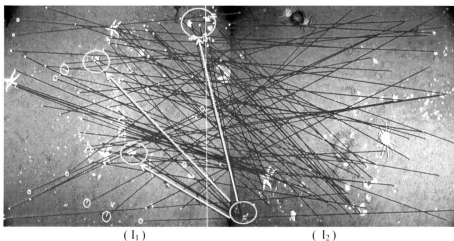


Figure 4.12: Illustrating features with multiple correspondences. SIFT Threshold = 0.66

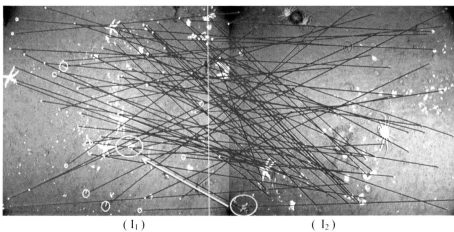


Figure 4.13: Multiple correspondence problem corrected. SIFT Threshold = 0.66

Here, snails in this pair of images are similar to a rotated replica of one another. Because of the invariance of SIFT to rotation the multiple matching problem in this case is expected. In order to solve this problem, an extra reverse correspondence finding stage was performed. Common members of the two stages are chosen as correspondences afterward.

#### 4.2.2.2.3 Feature Distance Metrics

The accuracy of the Nearest Neighbour (NN) classification for finding the distance between the feature descriptors significantly depends on the employed distance metrics. When there is no available prior knowledge about the descriptor vectors, the implementation of NN simply computes the Euclidean distances. Euclidean distance is a special case of the Minkowski distance metric. Equations 4.25 and 4.26 show the Euclidean and Minkowski metric consequently.

$$Euclidean = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (4.25)$$

$$Minkowski = \sqrt[r]{\sum_{i=1}^n |x_i - y_i|^r} \quad (4.26)$$

where  $x_i$  and  $y_i$  indicate the  $i^{th}$  element of vectors  $x$  and  $y$  with length  $n$ . For the special cases where  $r=1$ , the Minkowski metric shows the City Block metric,  $r=2$ , the Minkowski metric gives the Euclidean distance, and for  $r=\infty$  it gives the Chebychev distance.

The Minkowski distance family, and accordingly, the Euclidean distance, ignores any statistical regularity that might be estimated from the computing vectors [49]. In other words, Minkowski metrics do not take into account the difference of scales or dimensionality. For example, to classify images of faces by age and gender, it is not appropriate to use the same similarity metric for age and gender classification because their statistics presumably differ. This example attempts to emphasise the difference of scales or dimensions whilst computing distances between vectors. With the assumption that finding the similarity of a query photo of a person with a group of reference photos is demanded and that each photo has a descriptor vector containing two dimensions, sex and age.

$$\begin{array}{ccc} \underline{Person_1} & & \underline{Person_2} \\ sex_{p_1} & \leftrightarrow & sex_{p_2} \\ age_{p_1} & \leftrightarrow & age_{p_2} \end{array}$$

Assume the *age* variable range is 1-80 years old and the *sex* variable is 1 and 0 for *male* and *female* respectively. It can be clearly observed that the effect of *distance* between the sex variables will be overpowered by the age range in the condition where the selected *distance metric* treats the sex dimension similarly to the age dimension. In the following example a similar result will be achieved if the Minkowski metric is used, which does not take into account the statistical regularity of vectors.

$$\begin{array}{ccc} \underline{P_1} & & \underline{P_2} \\ sex: 0 & \leftrightarrow & 0 \\ age: 30 & \leftrightarrow & 31 \end{array} \quad \text{vs.} \quad \begin{array}{ccc} \underline{P_1} & & \underline{P_2} \\ sex: 0 & \leftrightarrow & 1 \\ age: 30 & \leftrightarrow & 30 \end{array}$$

$$\sqrt{(0-0)^2 + (30-31)^2} = 1 \quad \sqrt{(0-1)^2 + (30-30)^2} = 1$$

where 0 and 1 are associated with *female* and *male* variables respectively.

For matching SIFT descriptors, reviewed in section 4.2.1.1, we are coping with vectors of 128 dimensions as the formation of SIFT descriptors. Minkowski distance is appropriate for the situation where feature vectors are independent of each other and are of equal importance. This distance has been the most widely used measure for computing the similarity between feature descriptors. The Minkowski metric treats all the dimensions of the feature descriptors as entirely independent. In order to take into account the similarity of the descriptor vectors, the quadratic distance is introduced as shown in equation 4.27.

$$Dist(f_i, f_j) = \sqrt{(f_i - f_j) \Delta^{-1} (f_i - f_j)^T}, \quad (4.27)$$

where  $f_i$  and  $f_j$  are feature descriptors,  $\Delta$  is a similarity matrix. If  $\Delta = I$  ( $I$  is the identity matrix) equation 4.27 shows the Euclidean distance. In case  $\Delta = \Sigma$ ,  $Dist(f_i, f_j)$  gives the Mahalanobis distance, where  $\Sigma$  denotes the covariance matrix of the feature vectors. Also if  $\Delta = \Lambda$ , the equation denotes the standardized Euclidean distance where  $\Lambda$  is the diagonal of  $\Sigma$ .

The matrix of variance  $\Sigma$  is defined in equation 4.28 as well as  $\Lambda$  being defined in equation 4.29.

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ \sigma_{p1} & \cdots & \cdots & \sigma_{pp} \end{bmatrix}_{p \times p}, \quad (4.28)$$

$$\Lambda = \begin{bmatrix} \sigma_{11} & 0 & \cdots & 0 \\ 0 & \sigma_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{pp} \end{bmatrix}_{p \times p}. \quad (4.29)$$

Where  $N$  is the number of extracted keypoints in the dataset,  $f$  is the feature descriptor and  $p$  indicates the number of dimensions; the diagonal elements denote the covariance.  $\sigma_v$  is the covariance of  $f_i$  and  $f_j$ . Consequently,  $\sigma_v$  is the covariance of  $f_i$  with itself.

According to the formation of SIFT descriptors, 128 dimensions are not necessarily independent from each other. Therefore, using Minkowski metrics cannot be the best choice for finding the similarity between the SIFT features. Among quadratic-form metrics, Mahalanobis distance has been widely used in this area. The reason that the standardized Euclidean distance performs better than the Mahalanobis for us lies in the *grouping* of our images.

The possible scenarios for finding the matching keypoints in our situation could be categorized in two groups. Scenario one will describe the condition where there is prior knowledge available about the similarity of the field images and in contrast, scenario two describe the situation where prior knowledge about the collected images is not available. These two possibilities in our work are described further as below.

- Scenario #1: Group of images have very high similarity with each other.

The ultimate goal of this section is to determine if the descriptors of a query image has matches with any descriptors in the group of images taken from the field. In this scenario, firstly the covariance matrix  $\Sigma$  of all the reference images in the

training set should be computed. This is due to the high similarity of features in the field image. In other words, we already have some information about the *expected* and interesting keypoints in the set; therefore some *prior* information exists in this situation. Accordingly, using the covariance matrix is an appropriate choice. Employing the dependencies between dimensions in this computation is a reasonable choice due to the existence of high similarities in the dataset.

- Scenario #2: No prior knowledge about features in the images is available.

For example, we have a group of field images and the goal is to find similar features of a query image among the field image dataset. In this case, we are not able to firstly, categorize similar features and then analyse whether a query feature is similar to the group of features of the field image dataset or not. The solution for this situation depends on the possibly further available knowledge. The options could be as follows:

- (1) If dimensions of the feature vectors are not necessarily independent; therefore using Minkowski metric is inappropriate.
- (2) If there is no prior knowledge about features or the field image set, thus Mahalanobis may perform incapably. In this situation we suggest using the standardized Euclidean distance.

The standard Euclidean distance was previously introduced in equation 4.27. This distance metric basically applies standardization to balance out the contributions of variables in different scales of measurements; hence large variables will not dominate in



the calculation of distances and small variables will not be neglected. In this standardization, variables are transformed to have the same variance of one by centering the variables at their mean. This transformation is thus shown in equation 4.30.

$$\text{standardized value} = (x - \mu) / \delta, \quad (4.30)$$

where  $x$  is a sample element of our feature vector,  $\mu$  is the mean of the dimension of  $x$  and  $\delta$  denotes the standard derivation of the dimensions. In order to reformulate the quadratic distance for our purpose we have a discussion as follows:

We define data set  $P$  consisting of the SIFT feature descriptors  $S_p = \{S_{p_i}\}_{i=1}^{N_p}$  for a sample image being referred to previously as  $I_1$  and data set  $Q$  including descriptors  $S_q = \{S_{q_i}\}_{i=1}^{N_q}$  of a sample image  $I_2$ . The similarity function for a given pair of descriptors  $S_{p_i}$  and  $S_{q_j}$  from the data set  $P$  and  $Q$  is defined as the standardized Euclidean distance shown in equation 4.31.

$$Dist(p, q) = \sqrt{\sum_{k=1}^{128} (S_{p_{i,k}} - S_{q_{j,k}})^2 \Lambda^{-1}(S_{p_{i,k}} - S_{q_{j,k}})^2}. \quad (4.31)$$

Next, by using the *second-best match* approach for rejecting potential wrong matches using a distance ratio and a threshold, more of the potential incorrect matches will be discarded. In section 7.1 experimental results to support the idea of using standardized Euclidean distance for finding correspondences in one image pair will be presented as well.

#### 4.2.2.2.4 Match Finding Using the Adaptive Threshold

In this section, we present an approach for computing an adaptive threshold value in order to eliminate the demand for defining a constant pre-defined threshold value. We define the mean of the ratio of the first closest over the second closest distance as the adaptive threshold value as shown in equation 4.32.

$$\bar{t} = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{Dist(p_i, q_1)}{Dist(p_i, q_k)}, \quad (4.32)$$

where  $N_p$  denotes the number of features in  $I_1$ ,  $p_i$  is the  $i^{th}$  feature descriptor of set  $S_p = \{S_{p_i}\}_{i=1}^{N_p}$  and  $S_p$  is the  $i^{th}$  descriptor of  $I_1$ ,  $q_j$  is also the  $j^{th}$  descriptor of set  $S_q = \{S_{q_j}\}_{j=1}^{N_q}$ .  $Dist(p_i, q_j)$  shows the closest distance of descriptor vector  $p_i$  and  $q_j$ . The number of features in  $I_2$  is shown by  $N_q$ . The second closest distance for descriptor  $p_i$  and  $q_k$  is denoted by  $Dist(p_i, q_k)$ .

Euclidean and Mahalanobis distances are generally used to quantify the similarity of two feature vectors. Feature vectors contain 128-dimension digits and the components of the feature vectors are incomparable entities; therefore Euclidean distance is an inappropriate choice and yields an unsatisfactory outcome. The results to come in chapter 7.1 will illustrate the effect of the threshold value on the outcome of feature matching.

#### 4.2.2.2.5 Spatial Clustering

Clustering is widely used for identifying interesting patterns of data. The clustering problem is about categorizing the given data into groups called clusters. Using these clusters, characteristics of the dataset can be identified. Clustering has several

applications in pattern recognition, image processing, machine learning and market research, etc., [50]. K-means clustering [51] is one of the simplest unsupervised solutions for clustering problems. This algorithm classifies a given data set through a certain number of clusters. The main idea of this method is to associate  $k$  centroids for  $k$  clusters, one for each. These  $k$  centroids should be placed as far as possible from each other. The next stage in k-means is to associate the given data set members to the nearest centroid. The first stage is concluded when there is no point pending. Then  $k$  new centroids will be re-calculated by using the result of the previous step. The centroids are the mean point of the clusters. At this stage, the given data points will be re-associated to the new centroids. This process will be continued in a loop until centroids do not relocate. That is when there is no change in calculating the new centroids.

We have found that object description is possible with as few as three features to compute pose and location which is also mentioned in [34]. Using this fact, we calculate an initial value for  $k$  being used in k-means as shown in equation 4.33.

$$k = \frac{\text{number of features}}{n}, \quad n \geq 3, \quad (4.33)$$

where  $k$  is the number of clusters and  $n$  is the minimum number of features that can be used for object recognition. By using k-means clustering for corresponding pairs from the previous stage, we propose a technique to eliminate a large number of incorrect correspondences. The hypothesis is based on the fact that if a region of image  $I_1$  has overlap with a particular region in image  $I_2$ , the corresponding pairs inside these regions can merge into one cluster. In other words, if features inside one cluster in image  $I_1$  have

correspondences in several different feature point clusters in image  $I_2$ , this indicates incorrect correspondences. In contrast, if features inside one cluster in image  $I_1$  have correspondences in one cluster of image  $I_2$ , this situation indicates the correspondence pairs are more likely to be correct matches as illustrated in Figure 4.14 and Figure 4.15.

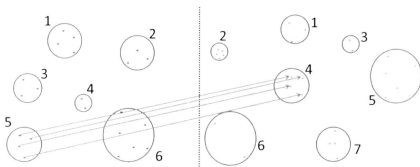


Figure 4.14: Illustrating an example situation where correspondences are considered as correct matches. Dots show feature points in each left and right images and circles illustrate clusters of feature points.

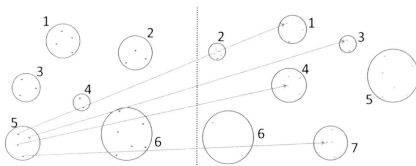


Figure 4.15: Features of one cluster in the left image have correspondences in several different feature point clusters in the right image; in this situation correspondent pairs are discarded.

Figure 4.16 and Figure 4.17 illustrate our strategy for rejecting correspondences using spatial clustering.

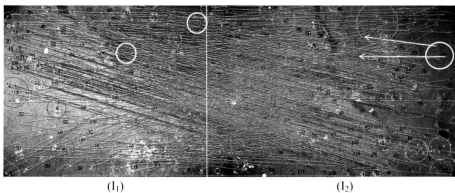


Figure 4.16: Keypoints of one cluster in  $I_2$  are associated with two clusters in  $I_1$ . This is the condition where the correspondences are rejected. Yellow arrows show the direction of two red lines.

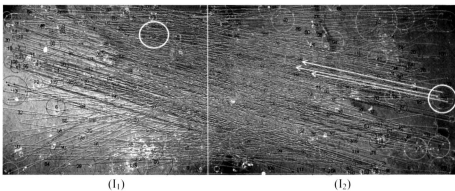


Figure 4.17: Illustrating the condition where correspondences are accepted. Yellow arrows show three lines between matched keypoints between a pair of clusters.

For the purpose of illustration, some sample clusters are highlighted in yellow circles for clarification. Two yellow arrows show the direction of the matched correspondences. Red is associated with rejected correspondences by clustering and blue shows accepted correspondences by this strategy. A large number of wrong matches of non-distinctive features, the sediments pattern herein, are filtered by using the proposed clustering technique.

# Chapter 5

## Image Registration

So far, a large number of corresponding features are obtained from the feature matching steps. In this chapter, the estimation of the geometric model between the camera viewpoints is discussed.

Image registration consists of establishing matches between a pair of images and then mapping the images into one plane. This requires finding the geometric transformation between the image planes.

### 5.1 Image Transformation Models

Most image registration models assume a planar scene and the rigid body motion of the camera. In this work, existence of the planar scene is our assumption as well. In addition, in the image database used in this research, the camera mounted on the submersible does not necessarily look downward.

An appropriate strategy to understand image transformation models is to break them into other simpler transformations. This is discussed in more detail in [52]. These transformations can be listed as rigid-body transformation, affine, projective or homographic and perspective transformations.

### 5.1.1 Rigid-Body Transformation (Isometric Transformation)

This geometric transformation preserves the distance between source points in an image and their correspondences in the mapped images. Accordingly angles in this transformation will be preserved. An isometry is basically a 2D rotation and a 2D translation adding up to three degrees of freedom. This transformation can be shown in equation 5.1:

$$p' = \begin{bmatrix} R & t_{2 \times 1} \\ 0^T & 1 \end{bmatrix} p, \quad (5.1)$$

where  $p = (x, y, 1)$  is defined as points in image  $I_1$  and  $p' = (x', y', 1)$  is the mapped points in image  $I_2$ .  $R_{2 \times 2}$  is the rotation matrix and  $t_{2 \times 1}$  denotes the translation vector.  $0^T$  is a  $2 \times 1$  zero matrix.

### 5.1.2 Similarity Transformation

This transformation is similar to an isometry except it includes a scaling factor invariant with respect to the direction (*Isotropic*). In this condition the distances are no longer invariant but angles are preserved. This transformation is shown in equation 5.2.

$$p' = \begin{bmatrix} sR & t_{2 \times 1} \\ 0^T & 1 \end{bmatrix} p, \quad (5.2)$$

where  $s$  is the scaling factor.



### 5.1.3 Affine Transformation

This transformation extends the similarity transformation. A similarity is composed of a rotation and an isotropic scaling factor whereas an affine transformation is composed of two rotations and two non-isotropic scaling parameters. These extra parameters add two degrees of freedom to a Similarity transform adding up to six degrees of freedom. Here, distances and angles are not preserved but the ratios of lengths of parallel lines are preserved. This translation is shown in equation 5.3 and 5.4.

$$P' = \begin{bmatrix} A & t_{2 \times 1} \\ 0^T & 1 \end{bmatrix} P, \quad (5.3)$$

$$A = R_\theta R_{-\phi} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} R_\varphi, \quad (5.4)$$

where  $R_\varphi$  and  $s_x$  are the rotation and scaling in the  $x$  axis.  $R_\theta$  and  $s_y$  are the rotation and scaling by the  $y$  direction respectively.

### 5.1.4 Projective Transformation or Homographies

Projective transformations include two more degrees of freedom than an affine transform. This will yield eight degrees of freedom. A homography can be written as equation 5.5.

$$P' = \begin{bmatrix} A & t_{2 \times 1} \\ V^T & v \end{bmatrix} P \quad (5.5)$$

The vector  $V$  distinguishes an affine transform from a homography. This vector is equal to zero in affine and is  $V = (V_1, V_2)$  in homography which is responsible for the projective effect of the transformation. In order to have a valid decomposition,  $v \neq 0$ .

A projective transformation can be shown as a composition of previously defined transformations shown in equation 5.6.

$$\begin{aligned}
 H &= H_{\text{Similarity}} \times H_{\text{Affinity}} \times H_{\text{Perspectivity}} = \begin{bmatrix} sR & t_{2 \times 1} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} U & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ V^T & v \end{bmatrix} \\
 &= \begin{bmatrix} A & t_{2 \times 1} \\ V^T & v \end{bmatrix}, \tag{5.6}
 \end{aligned}$$

where  $U = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$  and  $A = sRU + tV^T$ .

The homography between images  $i$  and  $j$  is denoted as  $H_{i,j}$ . According to the definition of the homography which is a  $3 \times 3$  matrix, only eight degrees of freedom exist. The plane itself has three degrees of freedom; the orientation of the camera has three degrees of freedom and finally the translation includes two degrees of freedom. A common solution for this situation is to divide the entire matrix element by the 9<sup>th</sup> element of the matrix.

### 5.1.5 Perspective Projection

So far the discussed transformation has dealt with 2D to 2D mapping. In the actual world condition, the images are transformed from 3D world space to 2D image points. This transformation can be represented by a perspective projection as shown in equation 5.7.

$$p' = \Gamma_{3 \times 4} p, \tag{5.7}$$

where  $p = (X, Y, Z, 1)^T$  is the actual world point represented in the homogeneous coordinates and  $p' = (x, y, 1)$  is the homogeneous coordinate of the mapped point to the

image plane. The projection matrix is a  $3 \times 4$  matrix which has eleven degrees of freedom up to an arbitrary scale. This matrix can be decomposed as shown in equation 5.8.

$$\Gamma = K[R|t], \quad (5.8)$$

where  $K$  is the camera's intrinsic matrix represented in equation 5.9 including five internal parameters.

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5.9)$$

Here,  $s$  is the skew parameter.  $\alpha_x$  and  $\alpha_y$  are the camera focal lengths in term of pixel dimension and  $(x_0, y_0)$  is the principal point of the image plane. The matrices  $R$  and  $t$  are the camera orientation and translation to the world including the six external parameters (three rotations and three translations) respectively.

As discussed in [52], some assumptions can be made to reduce the eleven degrees of freedom. If it is assumed the camera has square pixels, then  $\alpha_x = \alpha_y = \alpha$  and also in many cases  $s = 0$ . Using this assumption we will have nine degrees of freedom which is still one degree of freedom more than a homography.

In the condition where the intrinsic camera matrices are known, the best result can yield the extraction of three relative orientations of the camera as well as the image plane equation including three rotational parameters and three translational parameters. However, solving for the image plane equation is only possible if the camera intrinsic matrix is available as discussed in more detail in [53]. The intrinsic camera matrix was not available for our case. On the other hand, the homographic model exactly describes a

deformation of a flat scene photographed by a pin-hole camera, the optical axis of which is not perpendicular to the scene. An example of different motion parameters with two underwater vehicles is illustrated in Figure 5.1.

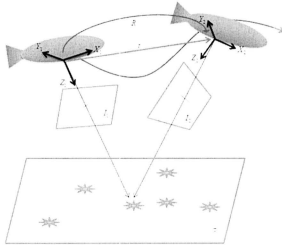


Figure 5.1: Submersibles pose: illustrating translation and rotation parameters affecting image frames and the overlap area.

Therefore, solving for a homography transformation will be the transformation of our choice. As the homography transform is written using the homogeneous coordinates, the homography  $H$  is defined using eight parameters plus a free 9<sup>th</sup> homogeneous scaling factor. Then for each corresponding points, we can obtain:

$$p' \approx Hp = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (5.10)$$

where  $H$  is the homography matrix and  $\approx$  indicates equality up to scale as  $h_{33} = 1$ . This equation can also be written as equations 5.11 and 5.12.

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{21}x + h_{22}y + h_{23}}, \quad (5.11)$$

$$y' = \frac{h_{31}x + h_{32}y + h_{33}}{h_{21}x + h_{22}y + h_{23}}, \quad (5.12)$$

Therefore, at least four point correspondences providing eight equations are required to estimate the homography.

## 5.2 Geometric Model Estimation

The RANdom SAmple Consensus (RANSAC) algorithm [48] is a general parameter estimation technique that is widely used in machine vision problems for reconciliation of the sample data with parameters of the known geometric model due to robustness and simple implementation. This sampling algorithm attempts to generate a solution by selecting the minimum number of data points required to estimate the desired model parameters. RANSAC uses the smallest possible set of data points to obtain an initial solution. We define inliers as corresponding points whose distribution fit a geometric model and for which outliers are correspondences which do not fit the geometric model. This model-fitting family can be categorized into three broad groups based on the trade-offs made between speed, robustness and accuracy [54]. This categorization is illustrated in Figure 5.2:

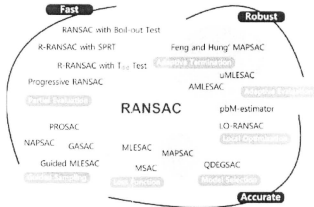


Figure 5.2: RANSAC family in 3 broad categories [54].

RANSAC is an iterative algorithm of two steps: firstly, hypothesis generation from random samples and secondly, the evaluation of the hypothesis with the data. It randomly selects a subset of data and estimates a parameter from the sample. If the parameters to be estimated fit the given model, the hypothesis is considered true. The RANSAC algorithm is summarized as follows:

---

#### RANSAC Algorithm

---

1. Randomly select the minimum number of required correspondence pairs to determine the model parameters.
2. Solve for the parameters of the model using correspondence pairs.
3. Determine how many correspondence pairs agree with the model with a predefined tolerance  $\epsilon$ . Inliers are defined as pairs which fit the model.

4. If the fraction of inliers over the total number of correspondences exceeds a certain threshold  $\tau$ , re-estimate the model parameters by considering all identified inliers and terminate.
  5. Otherwise, reiterate steps 1 through 4 to a maximum number of  $N$  times.
- 

The algorithm used here selects  $s = 4$  sample pairs from the pool of possible match correspondences [24]. The probability  $\rho$  is also the inlier ratio to the whole data. The number of iterations  $N$  is chosen high enough to ensure that at least one of the sets of random samples does not include outliers. However, the probability of selecting inliers is generally unknown; therefore, the number of iterations should be defined manually. In the hypothesis evaluation step, a pair of matched correspondences is recognized as the inlier candidate if its error from a hypothesis is less than a predefined threshold. The number of sample sets to be processed can be calculated as follows:

$\alpha$  : The estimated outlier proportion

$s$ : The size of the minimum required sample set of points

$\rho$ : The probability of finding a sample set of all inliers. We assume  $\rho = 0.99$ .

Therefore, we will have:

$1 - \alpha$ : Probability of a point being an inlier.

$(1 - \alpha)^s$ : Probability of a set of  $s$  points containing all inliers.

$1 - (1 - \alpha)^s$ : Probability of a set of  $s$  points containing an outlier.

$(1 - (1 - \alpha)^s)^N$ : Probability of  $N$  sets of size  $s$  samples all containing an outlier.

$1 - (1 - (1 - \alpha)^s)^N$ : Probability of  $N$  sets of points containing one outlier-free set.

Solving  $\rho = 1 - (1 - (1 - \alpha)^s)^N$  for the number of required iterations  $N$ , equation 5.13 will be obtained.

$$N = \frac{\log(1 - \rho)}{\log(1 - \alpha^s)} \quad (5.13)$$

Table 5.1 illustrates the required number of iterations for the different percentage of outliers in RANSAC compared to a determined linear system.

Table 5.1: Number of required iterations for geometric model estimation.

Sample size $s$	Proportion of outliers $\alpha$			Determined linear system 1000 data points
	0.1	0.5	0.7	
<i>Similarity</i> $\rightarrow 3$	4	35	167	$\binom{1000}{3} = 1.6 \times 10^8$
<i>Affinity</i> $\rightarrow 6$	7	293	6314	$\binom{1000}{6} = 1.3 \times 10^{15}$
<i>Homography</i> $\rightarrow 8$	9	1177	70188	$\binom{1000}{8} = 2.4 \times 10^{19}$

As can be seen in Table 5.1, the minimum number of required iterations will increase by increasing the number of sample points. Moreover, a dataset including a larger portion of outliers needs more iterations for the model estimation.



In our case, the objective of using RANSAC is to construct a robust estimation of the homography matrix. For each pair of images, the homography is estimated using at least four pairs of corresponding points. Practically, a larger number of correspondences could be employed to obtain an over determined linear system. By using equations 5.11 and 5.12 and rewriting  $H$  in a vector form as  $h = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^T$ ,  $n$  pairs of point-correspondences enable the construction of a  $2n \times 9$  linear system which is expressed in equation 5.14.

$$\begin{bmatrix} 0 & 0 & 0 & -X_1 & -Y_1 & -1 & y_1 X_1 & y_1 X_1 & y_1 \\ X_1 & Y_1 & 1 & 0 & 0 & 0 & -x_1 X_1 & -x_1 Y_1 & -x_1 \\ 0 & 0 & 0 & -X_2 & -Y_2 & -1 & y_2 X_2 & y_2 X_2 & y_2 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -x_2 X_2 & -x_2 Y_2 & -x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -X_n & -Y_n & -1 & y_n X_n & y_n X_n & y_n \\ X_n & Y_n & 1 & 0 & 0 & 0 & -x_n X_n & -x_n Y_n & -x_n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0_n \quad (5.14)$$

Solving this linear system involves the calculation called the Singular Value Decomposition (SVD). The SVD corresponds to rewriting the matrix in the form of the matrix product  $A = UDV^T$ , where the solution  $h$  corresponds to the last column of the matrix  $V$ ; then  $H$  is determined from  $h$  which is the solution for the parameters of the model in step 2 of RANSAC.

## Chapter 6

### Image Blending

In image formation, each pixel along a ray has a different intensity appearing in different images [32]. This issue appears more in underwater images as the submersible has to carry its own light source. This artificial light makes the center of the image brighter than the corners. An object for Mosaicing might appear once at the edge or corners of an image and another time in the center, having different intensity value. Another reason for different intensity levels in underwater images is the rapid attenuation of light in aquatic environments, which causes closer objects to have significantly higher intensity than further objects in a scene captured by camera. This change in intensity values makes the task of blending important for underwater images.

Once a pair of images is stitched together, the difference in the intensity level of the images can lead to clearly visible borders on the overlying area between the images. In order to solve this problem, a multiband blending approach is used. The method developed by Burt and Adelson [55] is performed for this stage. The idea behind multiband blending is that it decomposes images into several band-pass frequencies and then merges each frequency band rather than simply averaging the pixel's grayscale values. In this method, images for blending are firstly decomposed into different band-pass frequency components. Then each frequency band is merged separately by reassembling these frequency bands [32].

In order to generate the different band-pass frequency components, a Gaussian pyramid and Laplacian pyramid of images should be constructed. A Gaussian pyramid is formed by convolving the Gaussian low-pass filter with the original image  $G_0$ , followed by down sampling by a factor of two in order to obtain image  $G_1$ . As both the resolution and sample density are decreased from  $G_0$  to  $G_1$ , we can say  $G_0$  is the *reduced* version of  $G_1$  and also a REDUCE function is defined for this purpose as shown in equation 6.1.

$$G_i = REDUCE(G_{i+1}), \quad (6.1)$$

By applying the *REDUCE* function to the sequence of images  $G_0, G_1, \dots, G_n$  the Gaussian pyramid is constructed. Also *EXPAND* function is defined as an up sampler function by a factor of two in equation 6.2.

$$G_i = EXPAND(G_{i-1}), \quad (6.2)$$

The Laplacian pyramid corresponds to the different levels of the image frequency bands. This pyramid is constructed by taking the difference of levels in the Gaussian pyramid. This process is shown in equation 6.3.

$$L_i = G_i - EXPAND(G_{i+1}). \quad (6.3)$$

The Laplacian pyramids for each of the images are then added together in each level. The final seamless image can be reconstructed from the resultant Laplacian pyramids by inverting the process. The overall process is shown in Figure 6.1 for a 4-level pyramid.

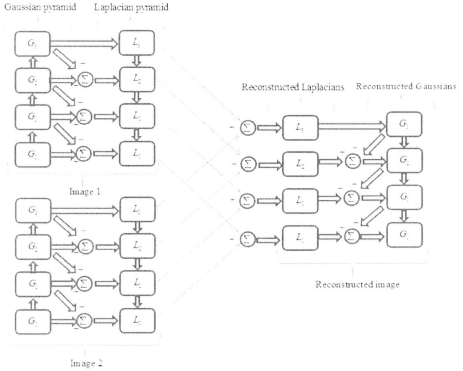


Figure 6.1: Multiband blending algorithm adds Laplacian pyramids of blending images and then reconstructs the seamless output image.

For the purpose of demonstration, an image from our dataset is divided into two images with different intensity levels so that the boundary between them is clearly visible. By applying the multi-band blending algorithm, we can clearly see that the boundary in Figure 6.2 is converted to a seamless transition of grayscale in Figure 6.3. This process significantly clarifies the appearance of the final mosaic by smoothing the sharp edges of each image in the mosaic.

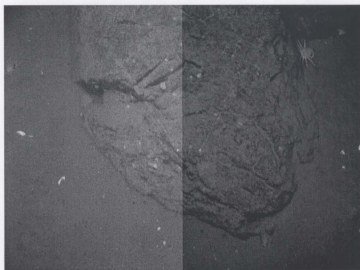


Figure 6.2: An image divided into two different intensity levels in order to generate a sample sharp edge between images.



Figure 6.3: A slight boundary is visible after applying the multi-band blending method.

# Chapter 7

## Results

In this chapter, we will show and compare the results obtained from the proposed strategy. The images used for this research have been collected by the Remotely Operated Platform for Ocean Sciences (ROPOS) [4] and the U.S. Geological Survey (USGS) [5]. Implementation of the program was coded in MATLAB using the VLFeat library [56] for extracting features and the Underwater Image Toolbox from [32] as well.

### 7.1 Distance Metrics Comparison

The purpose of a measure of similarity or distance is to compare two feature descriptors and compute a single number to evaluate this similarity. The following tables 7.1-7.3 show changes after decreasing the '*second best match threshold*' from 0.625 to 0.5 for several distance metrics. In this section we would like to show how matching is sensitive to the chosen threshold value to support our adaptive thresholding method. The threshold range used in Tables 7.1-7.3 is chosen manually so that the number of correspondences will be comparable. The SIFT feature extractor is controlled by two parameters, namely the *peak threshold* and the *edge threshold* as defined in [56]. The peak threshold eliminates peaks of the DoG scale space which are negligible and the edge threshold filters peaks of the DoG, of which the curvature is too small. Here, we chose a *peak threshold*=0 and an *edge threshold*=10 to detect as many keypoints as possible.

Throughout Tables 7.1-7.3 the correctness of matches is determined and counted manually.

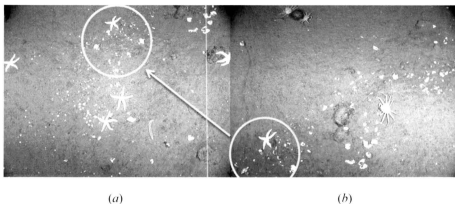


Figure 7.1: Sample image pair 1. The selected area shows the same region captured in two different photos.

As can be seen in Table 7.1, these distances are very sensitive to changes in the threshold. Moreover, not only the percentage of inliers over total correspondences is important, but in particular the number of inliers is crucial in order to estimate the homography matrix. The standardized Euclidean distance shows a more stable and more suitable result according to our application.

Parameters  $\alpha$  and  $\beta$  are defined as follows for clarity.

$\alpha =$  Number of total correspondences.

$\beta =$  Number of correct matches which are manually counted.

Table 7.1: Distance metrics and threshold value comparison #1.

Second-best match threshold	0.625		0.555		0.5		Average
Distance metric	$\frac{\beta}{\alpha}$	Ratio %	$\frac{\beta}{\alpha}$	Ratio %	$\frac{\beta}{\alpha}$	Ratio %	Ratio %
Euclidean distance	13/56	23	8/26	30	8/18	44	32
Standardized Euclidean distance	12/53	22	12/28	42	9/20	45	36
Mahalanobis distance	7/32	21	5/16	31	2/9	22	24
City block metric	13/54	24	10/31	32	9/21	42	32
Minkowski metric	12/56	21	9/26	34	8/18	44	33



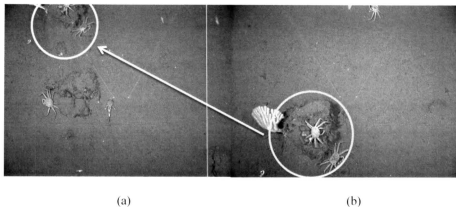


Figure 7.2: Sample image pair 2. The circled area shows the same region in two different photos.

Table 7.2: Distance metrics and threshold value comparison #2.

Second-best match threshold	0.625		0.555		0.5		Average
Distance metric	$\frac{\beta}{\alpha}$	Ratio %	$\frac{\beta}{\alpha}$	Ratio %	$\frac{\beta}{\alpha}$	Ratio %	Ratio %
Euclidean distance	16/40	40	12/24	50	9/15	60	50
Standardized Euclidean distance	17/40	42	12/22	54	10/17	58	51
Mahalanobis distance	8/20	40	6/13	46	6/9	66	50
City block metric	15/53	28	13/26	50	10/18	55	44
Minkowski metric	16/40	40	12/24	50	9/15	60	50

This attempt shows that if features in pairs of images are distinctive, the results of different distance metrics are similar. Table 7.3 will show the same analysis for a more challenging pair to support this idea. As can be seen, the average ratio for this condition is about 60%.

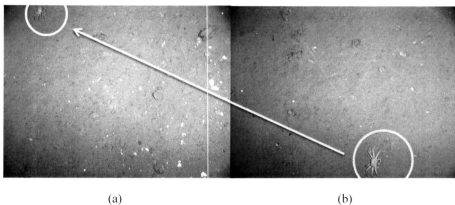


Figure 7.3: Sample image pair 3. The circled area shows the same region in two different photos.

Table 7.3: Distance metrics and threshold value comparison #3.

Second-best match threshold	0.625		0.555		0.5		Average
Distance metric	$\frac{\beta}{\alpha}$	Ratio %	$\frac{\beta}{\alpha}$	Ratio %	$\frac{\beta}{\alpha}$	Ratio %	Ratio %
Euclidean distance	26/50	52	14/24	58	11/15	73	61
Standardized Euclidean distance	26/47	55	14/23	60	7/11	63	59
Mahalanobis distance	14/19	73	6/9	66	4/7	57	65

City block metric	23/51	45	16/24	66	9/13	69	60
Minkowski metric	26/50	52	14/24	58	11/15	73	61

The tables in this section illustrate how the matching process is sensitive to the chosen threshold value ( $0.625-0.5$ ) and its influence on the number of correct matches for each metric; and moreover, slight changes in the threshold value can yield different results in finding the correct matches. The result of several distance metrics are also compared under variation of the threshold value. Considering the two factors of *higher average ratio* and *the number of correct matches  $\beta$* , standardized Euclidean distance shows a better performance compared to the other metrics.

## 7.2 Results of Feature Matching

To assess the matching rate, a pair of images with overlap are analysed by the conventional method and the proposed strategy. Figure 7.4 illustrates the pairs matched by the original algorithm and Figure 7.5 shows the matching pairs using the proposed approach (Section 4.2.2.2). By introducing the proposed approach, the matching performance is enhanced and there is a decrease in the false matching rate. Ultimately, the corresponding lines should be parallel in the absence of rotation following one single geometric model.

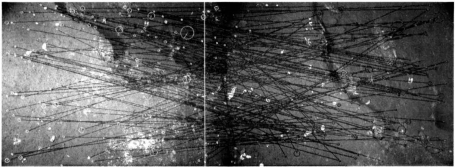


Figure 7.4: The conventional matching method, [34], with *threshold*=0.8. Resulting in 37 correct matches and 110 incorrect matches making up to 25% accuracy.

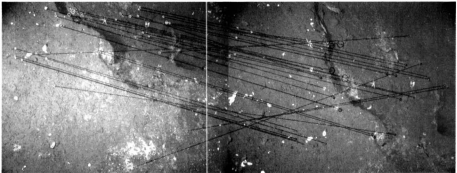


Figure 7.5: Feature matching using the proposed strategy. Up to 84% accuracy with 32 correct matches and 6 incorrect matches.

### 7.3 Final Image Mosaics

Figure 7.6 and Figure 7.7 show images which are stitched together for the purpose of solving the multiple counting problem. As an example, circled species in Figure 7.6 appeared in two different images previously causing a counting problem. By stitching these images together, we will have a single view of the investigated area of the sea floor; therefore, each animal will be seen once.

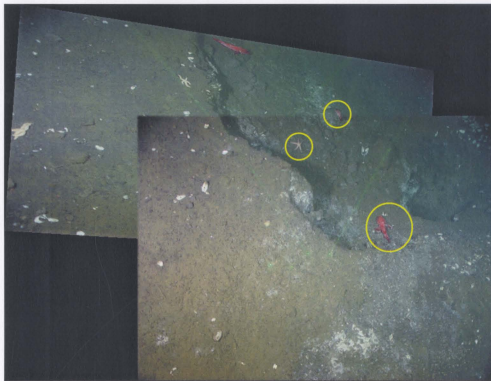


Figure 7.6: Mosaic created by stitching images with the multi-band blending function. The rockfish circled on the top was originally located on the boundary of one image; now it is clearly visible once in the image mosaic.

The same discussion is valid for Figure 7.7. This image can also be compared with the mosaic generated by the FFT-based method in Figure 4.3. Also in Figure 7.7, the upper corners of captured images could not be retrieved because of the lack of adequate lighting in the corner areas.



Figure 7.7: Mosaic of two images of starfish illustrating smooth boundaries multi-band blending algorithm. The starfish on top of the image is captured in two different photos appearing with minimum artifacts in the mosaic.

In order to produce a larger photo-mosaic, a database collected by the U.S. Geological Survey (USGS) [5] has been used. Figure 7.8 shows 6 images with enough overlap for mosaicing captured by a towed camera. Parameters shown in Table 7.4 are used for the USGS database presented in this chapter and also presented in Appendix A.

Table 7.4: Parameters of the implemented mosaicing algorithm.

Parameter	Value
SIFT edge threshold	10.0
SIFT peak threshold	0.0
RANSAC threshold	0.001
RANSAC iteration	1000
Number of sample points for RANSAC	8
Multiband blending pyramid size	3
Number of clusters	<u>Number of features</u> 9
Feature matching threshold	Adaptive

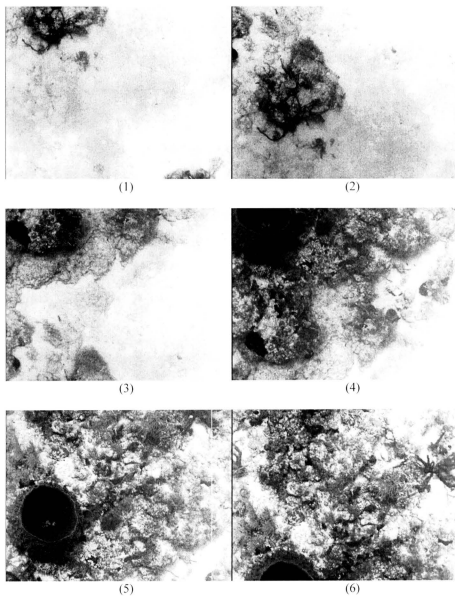


Figure 7.8: Image set #1, including images {1, 2, 3, 4, 5, 6}. Images are collected for mosaicing purpose by the U.S. Geological Survey.



In order to produce the photomosaic from images shown in Figure 7.8, each new image is stitched to the mosaic of the previous images by following the proposed method. We found that by using this method the mosaicing algorithm does not have to cope with accumulation errors. In the following mosaics, each new image (on the right-hand side) is matched with the mosaic of previous images (on the left-hand side). All correspondences are initially coloured red. Those which were accepted by condition of the clustering stage are coloured blue. Finally, correspondences which were selected by RANSAC as inliers are shown in green.

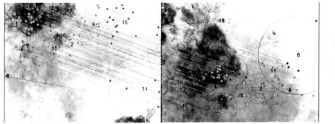


Figure 7.9: Feature matching between images  $\{1\}$  on the left and  $\{2\}$  on the right.



Figure 7.10: Images stitched  $\{1\} \leftarrow \{2\}$ .

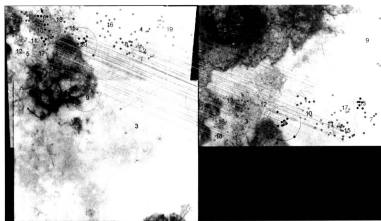


Figure 7.11: Feature matching between images  $\{1, 2\}$  on the left and  $\{3\}$  on the right.



Figure 7.12: Images stitched  $\{1, 2\} \leftarrow \{3\}$ .



Figure 7.13: Feature matching between images {1, 2, 3} on the left and {4} on the right.

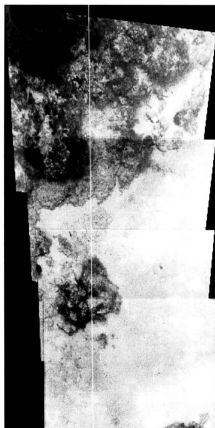


Figure 7.14: Images stitched  $\{1, 2, 3\} \leftarrow \{4\}$ .



Figure 7.15: Feature matching images {1, 2, 3, 4} on the left and {5} on the right.

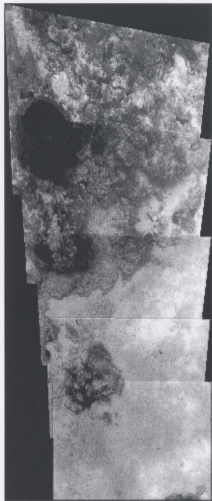


Figure 7.16: Images stitched  $\{1, 2, 3, 4\} \leftarrow \{5\}$  .



Figure 7.17: Feature matching between images {1, 2, 3, 4, 5} on the left and {6} on the right.

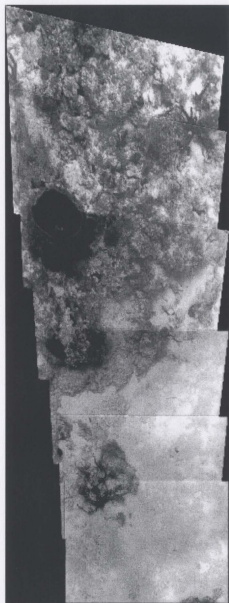






Figure 7.18: Images stitched  $\{1, 2, 3, 4, 5\} \leftarrow \{6\}$ . A photomosaic of 6 images from the sea floor collected by USGS.



Images 1 and 2 of the mosaic (as illustrated in Figure 7.8) are taken from a planar scene of the seafloor. Here we can see these two images are in a similar plane yielding a rectangular shaped mosaic. In the third image of Figure 7.8, some corals can be seen on the top left. We believe the ratio of the height of the coral to the distance between the camera and the scene has increased. This creates a semi-planar effect. Therefore, by mapping pixels from the top of image 3 to image 1's plane, a trapezoidal shaped mosaic will be formed. The semi-planar effect in images 4-6 does not change significantly. Consequently a similar shape of trapezoid could map image pixels into the first image plane. Varying threshold values for each step in producing the mosaic are shown in Table 7.5. The threshold shows an increasing trend when the mosaic enlarges. The reason is, since the mosaic enlarges, the size and number of features contained in the mosaic will increase. Consequently, the number of incorrect correspondences in the mosaic will increase. This increase in the number of incorrect correspondences will increase the mean value of the ratio accordingly.

Table 7.5: Adaptive thresholds computed by our proposed method for image set #1.

{Mosaic image} $\leftarrow$ New image	Adaptive computed threshold value
 {1} $\leftarrow$ {2}	0.7582
 {1,2} $\leftarrow$ {3}	0.7867
 {1,2,3} $\leftarrow$ {4}	0.7744
 {1,2,3,4} $\leftarrow$ {5}	0.8370

$\{1, 2, 3, 4, 5\} \leftarrow \{6\}$	0.8315
--------------------------------------	--------

This computation was implemented on a 32-bit MATLAB running on a 32-bit Windows operating system with 4GB of RAM. After stitching the sixth image of size  $680 \times 512$  pixels, we reached the memory limitations of MATLAB on this particular machine.

In order to illustrate the effectiveness of the presented feature matching strategy, a photomosaic of images in a group of six is created and included in appendix A along with tables of the adaptive thresholds for each image stitching stage.

#### 7.4 Discussion

In our attempt to solve the multiple counting problem, we performed several tests on different types of underwater images. Our imagery dataset was not intended to be used for image mosaicing when it was collected. This set includes images with the following characteristics:

- Images typically contain a high proportion of blue and green, making some details difficult to observe.
- High similarity of the sediment and shell patterns generates a large number of incorrect matches.
- Information about the camera calibration mounted on the submersible was unavailable.
- No accurate navigational data suitable for mosaicing purposes was available.
- The camera view angle was unknown.

These real conditions did not meet the constraints of the discussed mosaicing methods; therefore those methods failed to match many of the images in our database. Examples of the FFT-based registration and SIFT-based registration were presented and compared to illustrate the importance of the orthogonal view angle in Figure 4.3 and Figure 7.7 consequently. The importance of the camera viewpoint could be highlighted, according to the discussion about the FFT-based mosaicing on the one hand and scale and rotation invariance features of SIFT on the other hand. By performing several experiments using different distance metrics we could conclude the effectiveness of the standardized Euclidean distance for feature matching in the situation where training images are not available. In addition, by comparing changes in distances, we gained an understanding about the distinctiveness of features in an image. Image multiband blending performed satisfactorily, specifically in the case where an animal appears in the boundary of an image. Using this method we were able to construct the image in a way that the observed species appears clearly in the final mosaic.

# Chapter 8

## Conclusion

For solving the multiple counting problem for seafloor images, we aimed to design a system able to tolerate projective distortion and low contrast images. We found feature-based image registration methods are more applicable for the conditions. The designed system also had to cope with a large number of non-distinctive features from the background clutter.

In this thesis an improvement to the original SIFT feature matching algorithm was proposed. This improvement corresponds to enhancement of the feature matching algorithm in order to increase the percentage of the correct matches over the wrong correspondences. An adaptive threshold along with using spatial clustering in order to discard incorrect correspondences were also proposed. The presented mosaics illustrate the effectiveness of the proposed approach.

### 8.1 Future work

Throughout this thesis, we were using the term ‘distinctiveness’ of features in images. To the best of our knowledge, there is no parameter describing distinctiveness of features. In our feature matching stage, we investigate a situation which can be used for defining an index of distinctiveness in future; that is, if objects in an image *look easy to detect* by human eyes, e.g., comparing Figure 7.1 with Figure 7.3. The distance ratio of the

descriptor will have a small variance for non-distinctive features. On the other hand, we can see larger variance of the ratios for image pairs with distinctive features.

Even in a favourable situation, the camera mounted on a submersible will not necessarily be perpendicular to the seafloor due to the structure of the seabed itself. By having an estimate of the view angle and navigational data, an interesting approach for feature extraction could be designed consisting of simulation of the possible view angles. This is an approach similar to the viewpoint simulations in the Affine-SIFT [21], but with a limited required number of simulations. In this case the computational load of Affine-SIFT feature extraction will be reduced, but the overall registration process will benefit from more simulated viewpoints.

Marine biologists are also interested in investigating different types of seafloor in terms of coverage. For example, soft sediments are a desired habitat for crabs, and the bedrock area is where rockfish and starfish are mostly found. We believe that by analysing features of an image it will be possible to recognize different patterns on the seafloor. Thus, an automatic visual navigation system could be designed to navigate a submersible to search for a desired coverage on the seafloor.

### Bibliography

- [1] A. F. Gobi, "Towards Generalized Benthic Species Recognition and Quantification using Computer Vision," in *OCEANS 2010 IEEE*, Sydney , 2010, pp. 1 - 6.
- [2] D. Smith, M. Dunbabin, "Automated Counting of the Northern Pacific Sea Star in the Derwent Using Shape Recognition," in *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, 2007, pp. 500-507.
- [3] Nowak, B.M.; Whitney, T.; Ackley, S.F., "Analysis of ROV video imagery for krill identification and counting under Antarctic sea ice," in *Autonomous Underwater Vehicles, IEEE/OES*, Oct. 2008, pp. 1-9, 13-14.
- [4] K. Shepherd and S. Juniper., "ROPOS: "Creating a Scientific tool from an industrial ROV",," *Marine Technology Society Journal*, vol. 3, no. 31, pp. 48–54, 1997.
- [5] (2011, November) U.S. Geological Survey. [Online]. <http://www.usgs.gov/>
- [6] Edward E. Ruppert, Robert D. Barnes, *Invertebrate zoology* , 6th ed. Toronto : Fort Worth ; Saunders College Pub., 1994.
- [7] Love et al. , *The rockfishes of the northeast Pacific.*: University of California Press, 2002.
- [8] (2011, November) Centre for marine biodiversity. [Online]. <http://www.marinebiodiversity.ca/>
- [9] Kordelas, G.; Daras, P., "Robust SIFT-based feature matching using Kendall's rank correlation measure," in *16th IEEE International Conference on Image Processing*

- (*ICIP*), 2009, pp. 325-328.
- [10] Reddy, B.S., Chatterji, B.N., "A FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration," in *IEEE Transaction on Image Processing*, 1996.
- [11] R. Szeliski and S. Kang, "Direct Methods for Visual Scene Reconstruction," in *IEEE Workshop on Representations of Visual Scenes*, Cambridge, MA, 1995, pp. 26-33.
- [12] H. Shum and R. Szeliski., "Construction of panoramic mosaics with global and local alignment," in *International Journal of Computer Vision*, vol. 36(2), February 2000, pp. 101-130.
- [13] M. Brown and D. Lowe., "Automatic panoramic image stitching using invariant features," in *IJCV*, vol. 74(1), 2007, pp. 59-73.
- [14] A. Elibol, R. Garcia, O. Delaunoy, and N. Gracias, "A New Global Alignment Method for Feature Based Image Mosaicing," in *Proceedings of the 4th International Symposium on Advances in Visual Computing*, vol. 2, 2008, pp. 257-266.
- [15] Kudzinava, M., Garcia, R., Marti, J., "Feature-Based Matching of Underwater Images," in *International Workshop on Marine Technology*, 2007, pp. 96-97.
- [16] Naoki CHIBA, Hiroshi KANO, Michihiko MINOH and Masashi YASUDA, "Feature -based image mosaicng," in *IEICE*, vol. D-II , p. J82.
- [17] M. Brown and D. Lowe, "Recognising Panoramas," in *Proc. 9th Int'l. Conf.*

- Computer Vision*, 2003, pp. 1218-1227.
- [18] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey.," in *Foundations and Trends R in Computer Graphics and Vision*, vol. 3(3), 2008, pp. 177-280.
- [19] M. Brown and D. Lowe, "Automatic panoramic image stitching using invariant features," in *International Journal of Computer Vision*, vol. 74(1), 2007, pp. 59-73.
- [20] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," in *The International Journal of Robotics Research*, vol. 21(8), 2002. p. 735.
- [21] J.M. Morel and G.Yu, "ASIFT: A New Framework for Fully Affine Invariant Image Comparison," in *SIAM Journal on Imaging Sciences*, vol. 2, 2009.
- [22] Richard Szeliski, Heung Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* , 1997, pp. 251-258.
- [23] Marks, R.L.; Rock, S.M.; Lee, M.J., "Real-time video mosaicking of the ocean floor," in *IEEE Journal of Oceanic Engineering*, vol. 20, 2002, pp. 229 - 241.
- [24] W. H. WANG Yue, WU Yun-dong, "Free image registration and mosaicing based on tin and improved szeliski algorithm," in *ISPRS Congress*, Beijing, 2008.
- [25] YONGWEL SHENG ; PENG GONG ; BIGING Gregory S. , "True orthoimage production for forested areas from large-scale aerial photographs," in *Photogrammetric engineering and remote sensing*, vol. 69, 2003, pp. 259-266.



- [26] P. Pritchett and A. Zisserman, "Wide baseline stereo matching," in *ICCV'98*, 1998, pp. 754-760.
- [27] Deriche, Z. Zhang, Q. Luong, and O. Faugeras., "Robust recovery of the epipolar geometry for an uncalibrated stereo rig," in *ECCV'94*, 1994, pp. 567-576.
- [28] A. Baumberg, "Reliable feature matching across widely separated views," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 774-781.
- [29] Yudong Cao; Honggang Zhang; Yanyan Gao; Xiaojun Xu; Jun Guo, "Matching Image with Multiple Local Features," in *20th International Conference on Pattern Recognition (ICPR)*, Aug. 2010, pp. 519-522.
- [30] Wei Zhang; Kosecka, J., "Image Based Localization in Urban Environments," in *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, 2006, pp. 33 - 40.
- [31] Kashif Iqbal, Rosalina Abdul Salam, Azam Osman and Abdullah Zawawi Talib, "Underwater Image Enhancement Using an Integrated Colour Model," in *IAENG International Journal of Computer Science*, 2007, pp. 34:2.
- [32] R. Eustice, O. Pizarro, H. Singh, and J. Howland, "UWIT: Underwater Image Toolbox for Optical Image Processing and Mosaicing in MATLAB," in *Proc. IEEE Int'l Symp. Underwater Technology*, Apr. 2002, pp. 141-145.
- [33] R.Garcia, T.Nicosevici and X.Cu..., "On the way to solve lighting problems in underwater imaging," in *IEEE OCEANS Conference*, Biloxi Mississippi USA, 2002, pp. 1018-1024.

- [34] David G. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, vol. 2, 2004, pp. 91-110.
- [35] Oliver, K. ; Weilin Hou; Song Wang, "Feature matching in underwater environments using sparse linear combinations," in *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun 2010, pp. 60-67.
- [36] Daixian Zhu; Xiaohua Wang, "A Method of Improving SIFT Algorithm Matching Efficiency," in *2nd International Congress on Image and Signal Processing, 2009. CISP '09.*, Oct. 2009, pp. 17-19.
- [37] Ferrer, J.; Elibol, A.; Delaunoy, O.; Gracias, N.; Garcia, R., "Large-Area Photo-Mosaics Using Global Alignment and Navigation Data," in *OCEANS 2007* , 2007, pp. 1 - 9.
- [38] W. Zhang and J. Kosecka , "A new inlier identification scheme for robust estimation problems," in *Proceedings of Robotics: Science and Systems*, 2006.
- [39] Pablo d'Angelo, "Radiometric alignment and vignetting calibration ," in *The 5th International Conference on Computer Vision Systems*, Sep. 2007. [Online]. <http://hugin.sourceforge.net/>
- [40] K. Iqbal, R. Abdul Salam, A. Osman, and A. Zawawi Talib, "Underwater Image Enhancement Using an Integrated Colour Model," *IAENG International Journal of Computer Science*, vol. 34, p. 2, 2007.
- [41] Rafael C. Gonzales, Paul Wintz, *Digital image processing*, 2nd ed. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1987.

- [42] Nick. Efford, *Digital Image Processing: A Practical Introduction Using Java*. Essex: Addison-Wesley, 2000.
- [43] Barbara Zitova, Jan Flusser, "Image registration methods: a survey," *ELSEVIER Image and Vision Computing*, vol. 21, pp. 977–1000, June 2003.
- [44] Manjusha P. Deshmukh. Udhav Bhosle, "A SURVEY OF IMAGE REGISTRATION," *International Journal of Image Processing (IJIP)*, vol. 5, no. 3, 2011.
- [45] T. Suk J. Flusser, *IEEE Transactions on Remote Sensing*, pp. 382-387, 1994.
- [46] S. Tabbone D. Ziou, "Edge detection techniques - An overview," in *International Journal of Pattern Recognition and Image Analysis*, 1998.
- [47] C. Schmid K. Mikolajczyk, "A pformance evaluation of local descriptors," in *Proceedings of International Conference CVPR'03*, 2003.
- [48] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography," in *Communication Association and Computing Machine*, vol. 24(6), 1981, pp. 381-395.
- [49] Kilian Q. Weinberger , Lawrence K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," in *The Journal of Machine Learning Research*, 2009, pp. 207-244.
- [50] Chia-Hui Chang. and Zhi-Kai Ding, "Categorical data visualization and clustering," in *Data & Knowledge Engineering, Elsevier*, 2005, pp. 243–262.
- [51] J.B. MacQueen, "Some methods for classification and analysis of multivariate

- observations," in *Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.
- [52] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed.: Cambridge University Press, 2004.
- [53] O.D. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, no. 3, pp. 485–508, 1988.
- [54] S. Choi, T. Kim, and W. Y., "Performance evaluation of RANSAC family," in *20th British Machine Vision Conference*, 2009.
- [55] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics.," in *ACM Transactions on Graphics*, vol. 2(4), 1983, pp. 217–236.
- [56] Vedaldi, A., Fulkerson, B. (2008) VLFeat: An open and portable library of computer vision algorithms. [Online]. [www.vlfeat.org](http://www.vlfeat.org)
- [57] A. J. Lacey, N. Pinitkarn and N. A. Thacker, "An Evaluation of the Performance of RANSAC Algorithms for Stereo Camera Calibration," in *British Machine Vision Conference (BMVC)*, 2000.
- [58] L. Moisan, B. Stival, "A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix," in *International Journal of Computer Vision*, vol. 57:3, 2004, pp. 201-218.

# Appendix A

In this section four sets of images, each including six images collected by USGS, is mosaiced. This database includes 32 images. Images {1, 2, 3, 4, 5, 6} were mosaiced in section 7.3 . Due to the computational limitation, images are mosaiced in groups of six in the following manner:

Image set #1: {1, 2, 3, 4, 5, 6} presented in section 7.3,

Image set #2: {6, 7, 8, 9, 10, 11},

Image set #3: {11, 12, 13, 14, 15, 16},

Image set #4: {16, 17, 18, 19, 20, 21},

Image set #5: {21, 22, 23, 24, 25, 26},

Image set #6: {26, 27, 28, 29, 30, 31}.

Image number 32 is excluded from the mosaicing process due to not having overlap with the other images. Each set is followed by a table showing the adaptive threshold computed by our algorithm.

In this section the symbol  $\{x, y\} \leftarrow \{z\}$  is used to demonstrate the process of stitching image 'z' to the mosaic of images 'x' and 'y'.

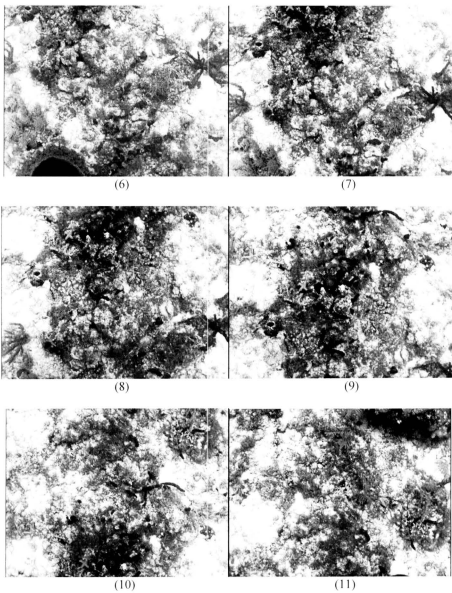


Figure A.1: Image set #2, including images {6, 7, 8, 9, 10, 11}.

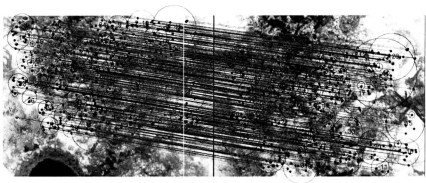


Figure A.2: Feature matching between images {6} on the left and {7} on the right.



Figure A.3: Images stitched  $\{6\} \leftarrow \{7\}$ .

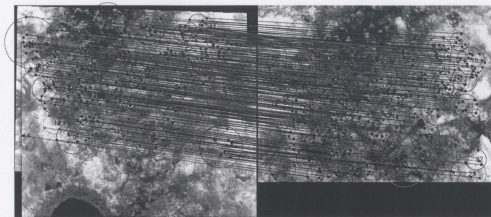


Figure A.4: Feature matching between images  $\{6, 7\}$  on the left and  $\{8\}$  on the right.

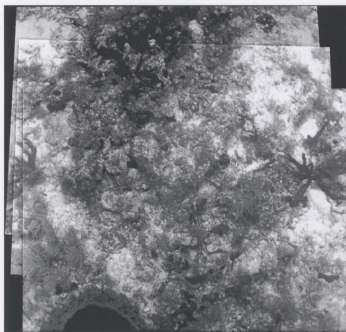


Figure A.5: Images stitched  $\{6, 7\} \leftarrow \{8\}$ .



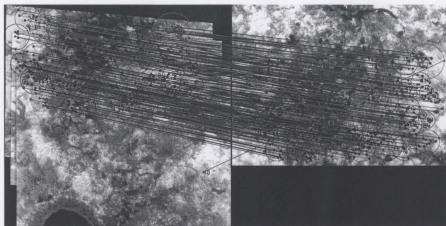


Figure A.6: Feature matching between images {6, 7, 8} on the left and {9} on the right.

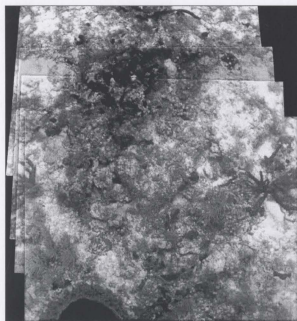


Figure A.7: Images stitched {6, 7, 8}  $\leftarrow$  {9}

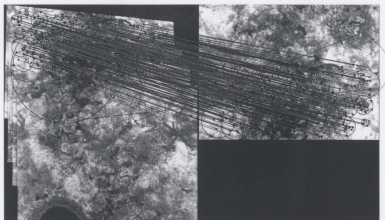


Figure A.8: Feature matching between images  $\{6, 7, 8, 9\}$  on the left and  $\{10\}$  on the right.



Figure A.9: Images stitched  $\{6, 7, 8, 9\} \leftarrow \{10\}$ .

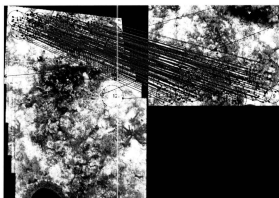


Figure A.10: Feature matching between images  $\{6, 7, 8, 9, 10\}$  on the left and  $\{11\}$  on the right.

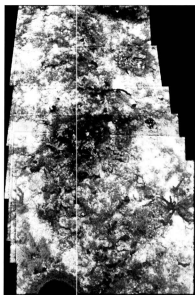


Figure A.11: Images stitched  $\{6, 7, 8, 9, 10\} \leftarrow \{11\}$

Table A.1: List of adaptive thresholds computed for image set #2.

$\{\text{Mosaic image}\} \leftarrow \text{New image}$	Adaptive computed threshold value
$\{6\} \leftarrow \{7\}$	0.6202
$\{6, 7\} \leftarrow \{8\}$	0.6347
$\{6, 7, 8\} \leftarrow \{9\}$	0.6885
$\{6, 7, 8, 9\} \leftarrow \{10\}$	0.7699
$\{6, 7, 8, 9, 10\} \leftarrow \{11\}$	0.8068

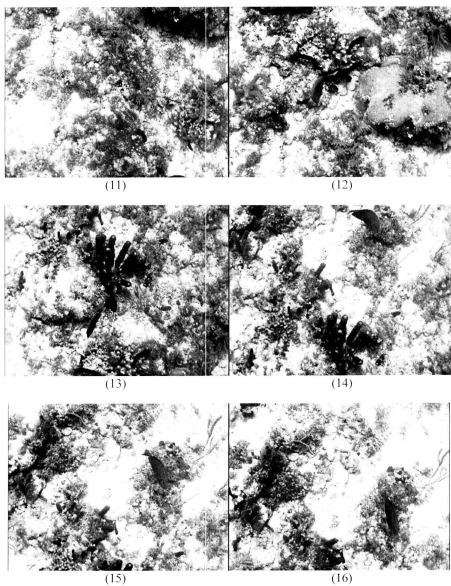


Figure A.12: Image set #3, including image {11, 12, 13, 14, 15, 16}.

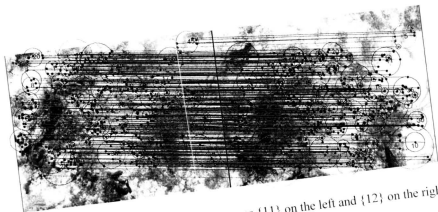


Figure A.13: Feature matching between images {11} on the left and {12} on the right.

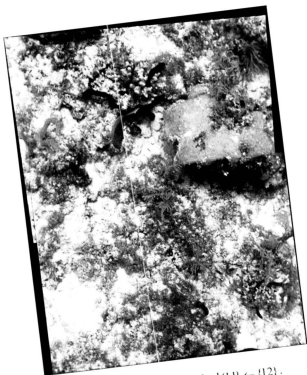


Figure A.14: Images stitched {11}  $\leftarrow$  {12}.

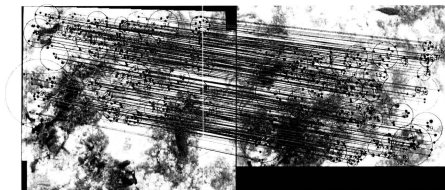


Figure A.15: Feature matching between images  $\{11, 12\}$  on the left and  $\{13\}$  on the right.



Figure A.16: Images stitched  $\{11,12\} \leftarrow \{13\}$ .

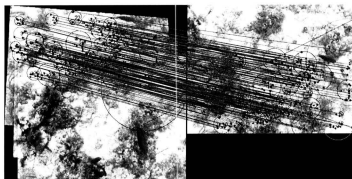


Figure A.17: Feature matching between images {11, 12, 13} on the left and {14} on the right.



Figure A.18: Images stitched {11, 12, 13} ← {14} .



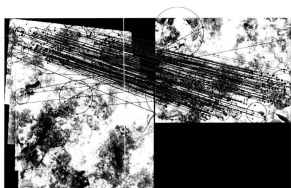


Figure A.19: Feature matching between images {11, 12, 13, 14} on the left and {15} on the right.



Figure A.20: Images stitched {11, 12, 13, 14} ← {15} .

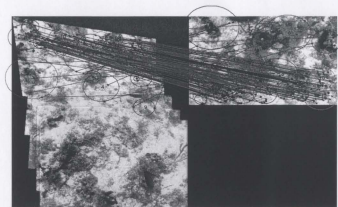


Figure A.21: Feature matching between images {11, 12, 13, 14, 15} on the left and {16} on the right.



Figure A.22: Images stitched {11,12,13,14,15} ← {16} .

Table A.2: List of adaptive thresholds computed for image set #3.

{Mosaic image} $\leftarrow$ New image	Adaptive computed threshold value
$\{11\} \leftarrow \{12\}$	0.7397
$\{11,12\} \leftarrow \{13\}$	0.7654
$\{11,12,13\} \leftarrow \{14\}$	0.6935
$\{11,12,13,14\} \leftarrow \{15\}$	0.7299
$\{11,12,13,14,15\} \leftarrow \{16\}$	0.7687

In this set of images, the blue fish appearing in images 12 and 13 has a relatively fast movement compared to the capture vehicle. Therefore, in this case the multiple counting problem is unavioded.

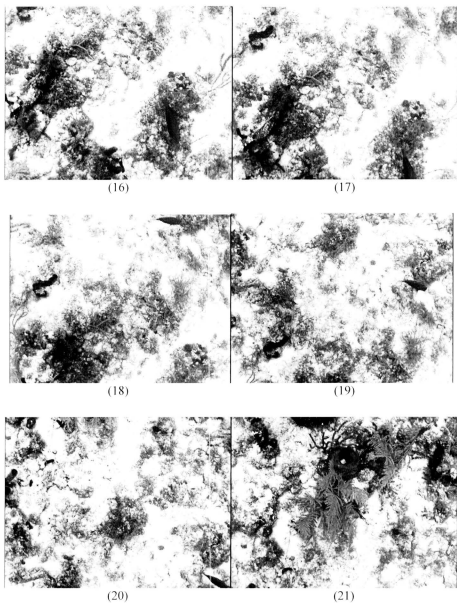


Figure A.23: Image set #4, including images {16, 17, 17, 19, 20, 21}.



Figure A.24: Feature matching between images {16} on the left and {17} on the right.

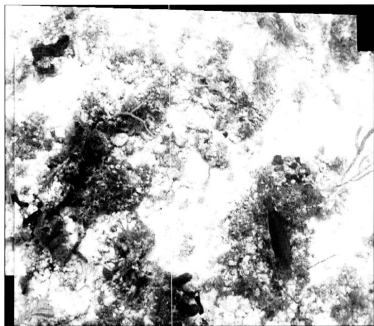


Figure A.25: Images stitched {16}  $\leftarrow$  {17}.

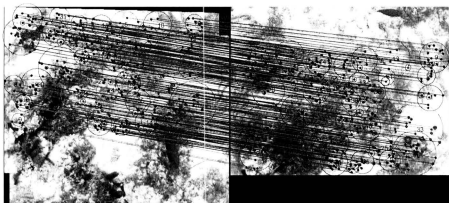


Figure A.26: Feature matching between images {16, 17} on the left and {18} on the right.

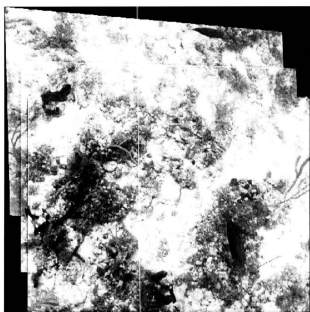


Figure A.27: Images stitched {16, 17}  $\leftarrow$  {18}.

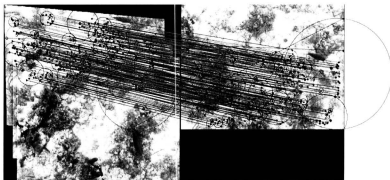


Figure A.28: Feature matching between images  $\{16, 17, 18\}$  on the left and  $\{19\}$  on the right.

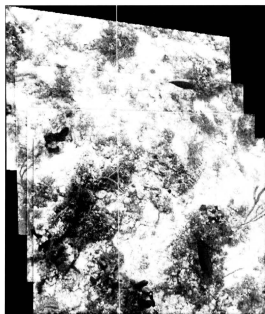


Figure A.29: Images stitched  $\{16,17,18\} \leftarrow \{19\}$ .

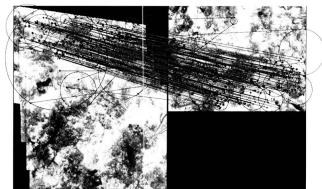


Figure A.30: Feature matching between images  $\{16, 17, 18, 19\}$  on the left and  $\{20\}$  on the right.

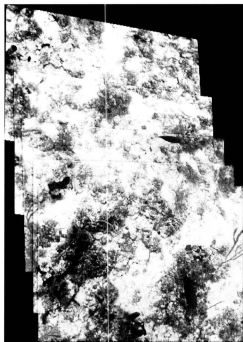


Figure A.31: Images stitched  $\{16,17,18,19\} \leftarrow \{20\}$ .



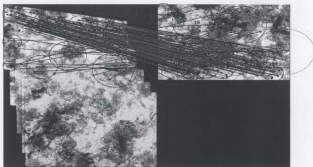


Figure A.32: Feature matching between images  $\{16, 17, 18, 19, 20\}$  on the left and  $\{21\}$  on the right.



Figure A.33: Images stitched  $\{16, 17, 18, 19, 20\} \leftarrow \{21\}$ .

Table A.3: List of adaptive thresholds computed for image set #4.

{ Mosaic image} $\leftarrow$ New image	Adaptive computed threshold value
{16} $\leftarrow$ {17}	0.6165
{16,17} $\leftarrow$ {18}	0.9657
{16,17,18} $\leftarrow$ {19}	0.7247
{16,17,18,19} $\leftarrow$ {20}	0.7958
{16,17,18,19,20} $\leftarrow$ {21}	0.8972

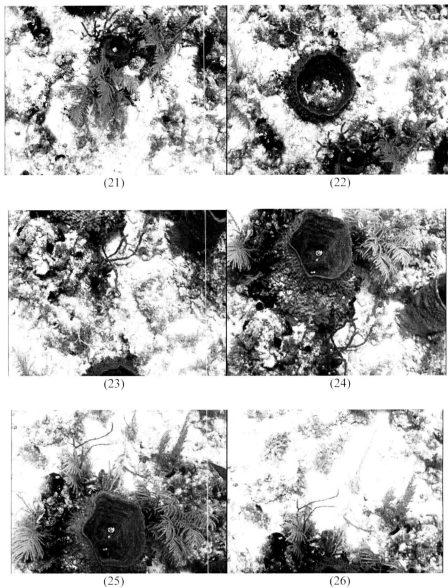


Figure A.34: Image set #5, including images {21, 22, 23, 24, 25, 26}.

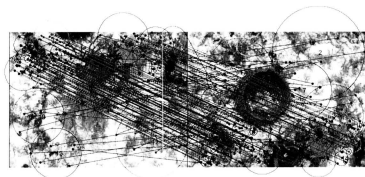


Figure A.35: Feature matching between images  $\{21\}$  on the left and  $\{22\}$  on the right.

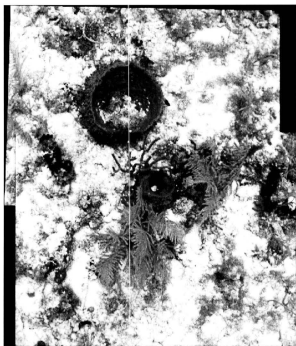


Figure A. 36: Images stitched  $\{21\} \leftarrow \{22\}$ .

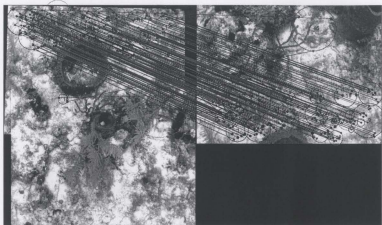


Figure A.37: Feature matching between images {21, 22} on the left and {23} on the right.

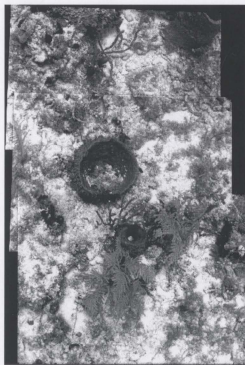


Figure A.38: Images stitched {21, 22} ← {23} .

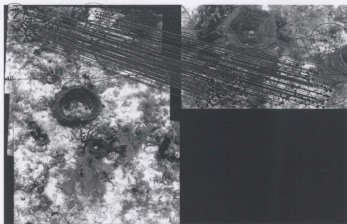


Figure A.39: Feature matching between images  $\{21, 22, 23\}$  on the left and  $\{24\}$  on the right.

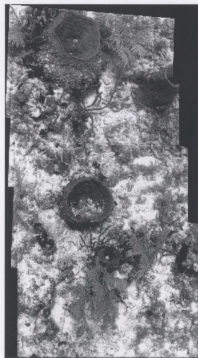


Figure A.40: Images stitched  $\{21, 22, 23\} \leftarrow \{24\}$ .

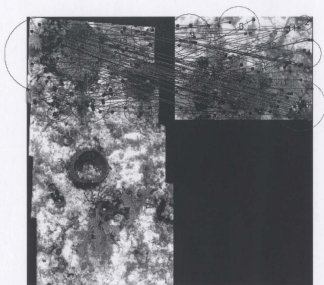


Figure A.41: Feature matching between images  $\{21, 22, 23, 24\}$  on the left and  $\{25\}$  on the right.



Figure A.42: Images stitched  $\{21, 22, 23, 24\} \leftarrow \{25\}$ .

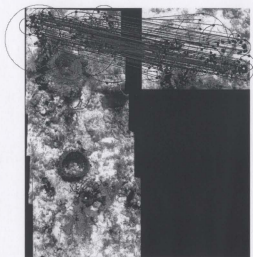


Figure A.43: Feature matching between images {21, 22, 23, 24, 25} on the left and {26} on the right.



Figure A.44: Images stitched {21, 22, 23, 24, 25} ← {26} .



Table A.4: List of adaptive thresholds computed for image set #5.

$\{\text{Mosaic image}\} \leftarrow \text{New image}$	Adaptive computed threshold value
$\{21\} \leftarrow \{22\}$	0.8324
$\{21, 22\} \leftarrow \{23\}$	0.7660
$\{21, 22, 23\} \leftarrow \{24\}$	0.7856
$\{21, 22, 23, 24\} \leftarrow \{25\}$	0.8817
$\{21, 22, 23, 24, 25\} \leftarrow \{26\}$	0.8727

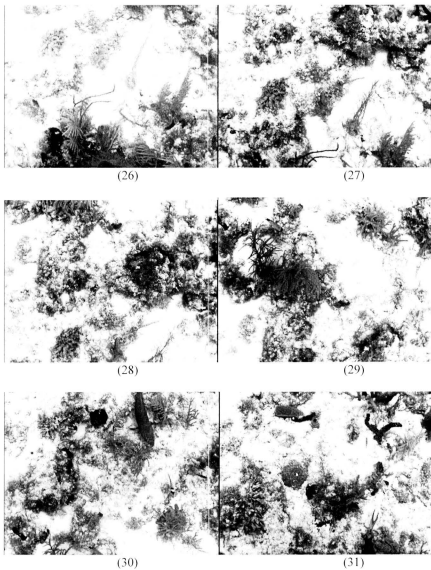


Figure A.45: Image set #6, including images {26, 27, 28, 29, 30, 31}.

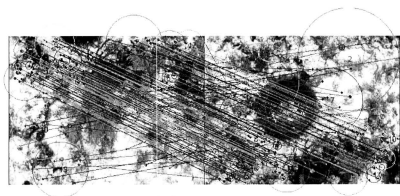


Figure A.46: Feature matching between images {26} on the left and {27} on the right.



Figure A.47: Images stitched {26} ← {27}.

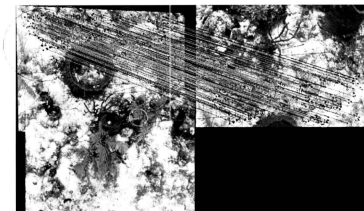


Figure A.48: Feature matching between images {26, 27} on the left and {28} on the right.

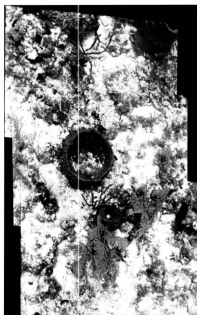


Figure A.49: Images stitched {26, 27} ← {28}.

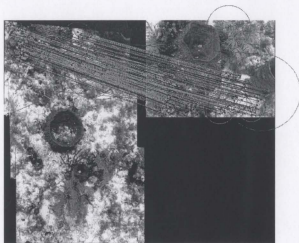


Figure A.50: Feature matching between images {26, 27, 28} on the left and {29} on the right.

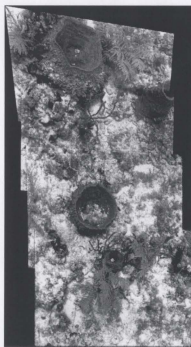


Figure A.51: Images stitched {26, 27, 28}  $\leftarrow$  {29}.

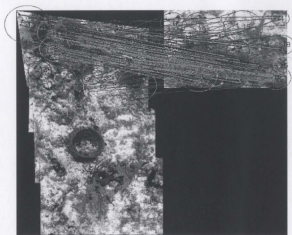


Figure A.52: Feature matching between images {26, 27, 28, 29} on the left and {30} on the right.

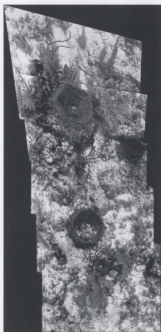


Figure A.53: Images stitched {26, 27, 28, 29} ← {30}.

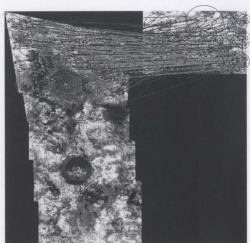


Figure A.54: Feature matching between images {26, 27, 28, 29, 30} on the left and {31} on the right.



Figure A.55: Images stitched {26, 27, 28, 29, 30} ← {31} .

Table A.5: List of adaptive thresholds computed for image set #6.

$\{\text{Mosaic image}\} \leftarrow \text{New image}$	Adaptive computed threshold value
$\{26\} \leftarrow \{27\}$	0.7860
$\{26, 27\} \leftarrow \{28\}$	0.7497
$\{26, 27, 28\} \leftarrow \{29\}$	0.7780
$\{26, 27, 28, 29\} \leftarrow \{30\}$	0.8654
$\{26, 27, 28, 29, 30\} \leftarrow \{31\}$	0.8924









